DOI: 10.21917/ijsc.2022.0359

GENE BICLUSTERING ON LARGE DATASETS USING FUZZY C-MEANS CLUSTERING

M. Ramkumar[1], J. Gowrishankar[2], V. Amirtha Preeya[3], T. Pushpa[4] and T. Karthikeyan[5]

[1,4]*Department of Computer Science and Engineering, HKBK College of Engineering, India*

[2]*Department of Computer Science and Engineering, Jain University, India*

[3]*Department of Computer Science and Engineering, Presidency University, India*

[5]*Department of Electronics and Telecommunications Engineering, University of Technology and Applied Sciences, Oman*

Abstract

The current study employs biclustering to alleviate some of the drawbacks associated with gene expression data grouping. Different biclustering algorithms are used in this study to detect unique gene activity in various contexts and reduce the duplication of broad gene information. Furthermore, machine learning or heuristic algorithms have become widely utilised for biclustering due to their suitability in problems where populations of potential solutions allow examination of a larger percentage of the research area. To begin with, gene expression data biclusters frequently contain data that is the same under a variety of different situations of gene expression. Therefore, the biclustering technique is particularly effective if the matrix lines and columns are merged immediately. Submatrices can be identified using the Large Average Sub matrix. A Fuzzy C-Means algorithm is also used to ensure that the sub-matrix can be expanded to include more rows and columns for further analysis. The sub-matrices and component precision and strength are factored into the system design. It uses biclustering techniques to differentiate gene expression information. On the Garber dataset, the simulation is run in Java. Using the average match score for non-overlapping modules, the influence of noise on overlapping modules using constant bicluster and additive bicluster, and the overall run duration, the study is assessed.

Keywords:

Heuristic Algorithm, Gene Expression, Data Biclusters, Fuzzy C-Means

1. INTRODUCTION

DNA microarray technologies help to measure levels of expression in experimental circumstances of thousands of genes [1]. Local patterns have motivated the large study to use pattern-based searches to deal with them. Due to its capacity to uncover hidden designs, the use of biclustering in biological data is common.

In the field analysis of gene expression data, in particular, biclustering is very important. Its primary objective is to be able to identify groups of genes that act equally under a subset of samples (conditions). But the pioneering literature algorithms have shown certain limitations on the quality of biclusters that were unveiled.

A network of biological entities, e.g., genes, proteins, metabolists and so on, is linked together [2]. One of the key issues in bioinformatics is analyzing and extracting biologically significant information from these entities.

The mechanism for generating a protein from the gene is gene expression. In two main steps, transcript and translation, this process happens. While transcription involves the production and processing of the resultant m RNA molecule by the enzyme RNA polymerase, the translation step requires use of mRNA in direct protein synthesis and post translation of the molecule.

The concentration of mRNA in numeric values, namely gene expression information is measured using microarray DNA technologies. These technologies, known as DNA micro-array technologies, enable the evaluation in various experimental conditions of expression levels of thousands of genes [3]. Indeed, for numerous biologists, these technologies have become indispensable tools. This is because genomes are used to control broad levels of gene expression in a particular organism. A microarray is usually a glass slide to which DNA molecules are orderly fixed in certain locations (or features) called spots.

	$Condition_1$	…	$Condition_1$	…	$Condition_m$
$Gene_1$	m_{11}	…	m_{1j}	…	m_{1m}
⋮	⋮	…	⋮	…	⋮
$Gene_i$	m_{i1}	…	m_{ij}	…	m_{im}
⋮	⋮	…	⋮	…	⋮
$Gene_n$	m_{n1}	…	m_{nj}	…	m_{nm}

Fig.1. Gene expression data matrix

DNA micro-array was used in various areas of research, including gene discovery [4], the diagnosis of disease [5] and drug findings [6]. Microarrays are used to identify the functions of the genes and the mechanisms that underlie diseases.

For this purpose, data on gene expression are placed in a data matrix (Fig.1). Genes, and columns, experimental conditions, represent the samples, and every matrix cell denotes a gene level of expression in a certain experimental situation. In this context, it is of paramount significance to detect transcriptional gene modules co-regulated in a set of experiments [7].

Text mining is the technology that semi-automatically detects patterns and trends from large collections. It builds on a number of technologies, such as the processing of natural languages, information recovery, information extraction, and data mining [8]. There is currently a great deal of work being done in this field, mainly because the literature has exponentially increased biological knowledge in order to find useful and required data from a large variety of resources.

In addition, biological information is provided online in a combination of various forms, including structured, semi-structured and unstructured forms, making the use of computer techniques indispensable for this task. The authors in [10] has written an examination of various approaches to text mining for biological data.

Some applications based on the integration of various approaches include those of authors in [11], which demonstrate

that the results are improving significantly by combining different text mining algorithms with their results. In addition, a system called CONAN integrating various processes and biological information such as tagging gene or protein names, finding interaction and mutation information, tagging biological concepts and linking them to MeSH and Gene Ontology terms is proposed.

Recently, the authors in [12] have introduced a range of visualizing tools to navigate biomedical records and concepts efficiently and efficiently. Their approach is called bioTextQuest, combining automated discovery of significant terms with structured knowledge annotation in article clusters.

2. BACKGROUND

In order to achieve new and relevant patterns and facts, the data mining is a step in the KDD process that corresponds to applying specific algorithms for analysis of previously pre-processed data. The pattern is extracted means that a model is matched to a dataset, that the structure of the data is found or that a large data collection is described in general [13].

The original data are modified in various manner during the selection, processing and transformation steps to reduce its dimensionality or noise, while simultaneously trying to minimize the loss of the relevant information. An expert, although with the support of some computer tools, usually evaluates and interpret the results of the data mining process. In biological data the issue is how to bridge the two fields KDD and BI to discover sequence patterns, gene function, protein-protein interactions or any other useful data-dependent knowledge successfully [14].

Although every step in the KDD process is just as important as data mining, the KDD data mining component, which receives the greatest literature attention, will be our focus. The component for the data mining of KDD is strongly based on the known techniques used to learn machines, recognize patterns and find data patterns [15].

The verification and discovery process can be used to achieve two different objectives. The first part of a user hypothesis had to be examined and the second part sought to find new patterns in the data. This finding can be subdivided into a forecast, where the patterns obtained are utilized for predicting behavior of new entities, and a description of the patterns used to categorize and present the subjects studied. Discovery-oriented data mining is mostly used to study and analyze biological data, although verification-oriented techniques are normally used for validation procedures.

2.1 BICLUSTERING PROBLEM

Biclustering Problem is of utmost importance to discover transcriptional modules of genes which are co-regulated through a series of experiments.

Of course, in many challenges in bioinformatics, the clustering technology was shown to be beneficial. In fact, researchers can collect data such as cancer, certain sub-types of tumors and cancer survival rates. While the results were encouraging, clustering algorithms were used. There are two main disadvantages to clustering algorithms:

- They take the entire set of samples into account. This is despite the fact that not all samples are subject to genes. They can instead only be applicable to a subset of samples, which are a key aspect of many problems in the field of biomedicine. Therefore, both genes and conditions should be clustered simultaneously.
- Only in one group can each gene be clustered. However, many genes can be included in various clusters according to their effect in various biochemical processes.

Biclustering, which is one type of clustering, has palliated these inconveniences in this respect. Biclustration thus aims to identify maximum submatrices (along with biclusters) where a subset of genes is highly correlated with a variety of conditions. However, biclustering is a highly combinatorial and NP-hard problem.

Biclustering use is common in the analysis of gene expression data, as can be seen in a dedicated literature. Below we remember some basic definitions taken from the field of biclustering.

Definition 2.1. (Bicluster): A bicluster is a subset of objects (genes) associated with a subset of attributes (conditions) in which rows are co-expressed.

The bicluster associated with the matrix $M=(I,J)$ is a couple (A,B), such that $A \subseteq I$ and $B \subseteq J$, and (A,B) is maximal if there does not exist a bicluster (C,D) with $A \subseteq C$ or $B \subseteq D$. This leads us to the definition of biclustering.

Definition 2.2. (Biclustering): The biclustering problem focuses on the identification of the best biclusters of a given dataset. The best bicluster must fulfill a number of specific homogeneity and significance criteria (guaranteed through the use of a function to guide the search).

3. FCM ALGORITHM

A clustering technique that is separated from hard k-means using hard partitioning (Fuzzy C-means clustering, FCM) is also known as Fuzzy ISODATA. The FCM uses fuzzy partitions so that all groups with different membership levels of 0 to 1 can have a data point.

FCM is an algorithm iterative. FCM aims to identify cluster centers (centroids) that minimize differences. The membership matrix (U) will be randomly initialized according to Eq.(1) to support the introduction of fuzzy partitioning.

$$\sum_{i=1}^{c} u_{ij} = 1, \forall j = 1, ..., n \tag{1}$$

Eq.(2) gives the differential function used for FCM.

$$J(U, c_1, c_2, ..., c_c) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \sum_{j=1}^{n} {u_{ij}}^m {d_{ij}}^2 \tag{2}$$

where, u_{ij} is between 0 and 1; c_i is the centroid of cluster i; d_{ij} is the Euclidian distance between i^{th} centroid (c_i) and j^{th} data point; $m \in [1,\infty]$ is a weighting exponent.

To reach a minimum of dissimilarity function there are two conditions. These are given in Eq.(3) and Eq.(4).

$$c_i = \frac{\sum_{j=1}^{n} {u_{ij}}^m x_j}{\sum_{j=1}^{n} {u_{ij}}^m} \tag{3}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{d_{ij}}{d_{kj}}\right)^{2/(m-1)}} \quad (4)$$

Detailed algorithm of fuzzy c-means determines the following steps.

Step 1: Randomly initialize the membership matrix (U) that has constraints in Eq.(1).

Step 2: Calculate centroids (c_i)

Step 3: Compute dissimilarity between centroids and data points using Eq.(2).

Step 4: Stop if its improvement over previous iteration is below a threshold.

Step 5: Compute a new U using Eq.(4). Go to Step 2.

By iteratively updating the cluster centers and the membership ratings for each data point, FCM moves the cluster centers in a data set to the correct place. FCM does not make sure that the optimal solution converges. The use of U is initialized at random because of cluster centers using Eq.(3).

Initial centroids depend on performance. Two methods are described below for a robust approach: 1) To determine all centroids using an algorithm. 2) Run FCM with different initial centroids, multiple times each.

3.1 FITNESS FUNCTION

Having taken into account the various objectives of the biclustering assessment, we present here how they were combined to provide the fitness function for the evaluation of potential solutions in our algorithm. Although we have used the above-described four objectives in our experiments, the fitness function can be easily configured by adding a new mathematical formula.

The fitness function is in the context of evolutionary algorithms a specific type of objective function that is used to sum up how close a design solution can be to achieve the set targets as a single merit figure.

The final fitness function of our algorithm is shown in Eq.(5). The objective is to minimize the value of each term so that large biclusters with a low Transposed Virtual Error or TVE value, a high gene variance and a little overlap are found. The aim is to minimize the value of every word.

$$\Phi_B = \frac{VE^T(B)}{VE^T(M)} + w_s Vol(B) + w_{ov} o(B) + w_{\text{var}} \frac{1}{1+GV(B)} \quad (5)$$

All terms are weighted, except for TVE as the benchmark. The value of TVE was, however, divided into the TVE value of the whole microarray for the biclustering. This is because the range of TVE values in each microarray depends on the values. Although the algorithm tries to minimize it, when using a different microarray the weight of the other terms of fitness function must be reconstituted. To avoid this situation, we divide the whole microarray with the TVE value (M refers to the data matrix for the microarray).

The algorithm leads to different types of biclusters according to their sizes, overlap or variance between genes, changing the weights associated with different goals. All weights have been laid down the same way; biclusters with lower values for the relevant characteristic are lowered by a certain weight and vice versa.

In the results section, we provide default values for each weight that were obtained on an experimental basis and produced significant results for all the studied databases. We also offer guidance to the user on how the weight changes affect the various characteristics of the biclusters obtained.

Note that adding new fitness goals is quite simple. For each new biclustering feature, a new mathematical formula should be designed. When inserted in the fitness function, this formula will be minimized and its weight is also appropriate. It is preferable to set the range of values not on the specific values of the microarray or bicluster in order to better monitor the effect on the results.

3.2 FCM BICLUSTERING ALGORITHM

The proposed system uses FCM to check the number of columns and rows in the sub-matrix for calculation. The conventional cluster algorithms tend to attribute data to a cluster without considering how large a cluster is. However, the flush clustering has established a degree of membership that enables every data point to be made up of several clusters with a different degree of membership.

Every cluster is represented by the parameter vector that oscillates in FCM algorithm θ_j where j = 1,2,…,c and c is the total number of clusters.In FCM, the assumption is that a data point from the X dataset does not belong exclusively to a group, but can be part of more than one cluster at a certain degree simultaneously. The u_{ij} variable represents an x_i membership level in the C_j cluster.

The data point is more susceptible to the cluster with a higher membership value. In all clusters of a given data point, the total membership value is considered as

- An additional parameter called fuzzifier q (≥1) (fuzzifier) is used for the algorithm. The value preferable of the fuzzifier unit is considered as
- However, then the study observes the difference with various other values. The higher the q value is, the lesser is the generalization of the FCM algorithm.

FCM algorithm is formed from the cost minimization function, which is expressed as follows:

$$J(\theta,U) = \sum_{i=1}^{c}\sum_{j=1}^{c} u_{ij}^{q} \left\| x_i - \theta_j \right\|^2 \quad (6)$$

The FCM algorithm is considered the most popular algorithm, which is regarded as an iterative process. In the process of iteration, the following are important steps:

The membership degree, u_{ij} of an image x_j is represented in terms of a cluster C_j, i=1,2,…,N, and j=1,2,…,c, which is computed using Euclidean distance of x_i over all θ_j'.

$$u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{d(\vec{x}_i,\vec{\theta}_j)}{d(\vec{x}_i,\vec{\theta}_k)}\right)^{\left(\frac{1}{q-1}\right)}} \quad (6)$$

Then θ_j is considered as a representative, which is updated at regular instance based on the weighted means of the image vectors.

$$\theta_j = \frac{\sum_{i=1}^{n} \left(u_{ij}\right)^q \vec{x}_i}{\sum_{i=1}^{n} \left(u_{ij}\right)^q} \quad (7)$$

A number of methods can be used to terminate the FCM algorithm. The algorithm can be terminated if there is a small difference in values between θ_j or membership grade between two successive iterations. However, there are predetermined numbers of iterations.

The FCM algorithm is considered to be sensitive during the outlier presence, since its requirement is based on following expression:

$$\sum_{j=1}^{m} u_{ij} = 1 \quad (8)$$

The Eq.(8) is used to represent the noise point, which is to be accounted in order to acquire higher membership grade in a cluster.

4. RESULTS AND DISCUSSIONS

The Table.1 shows the parametric settings of different methods. The parametric setting used for the proposed method includes S=3, K=25 and N_bB_i=10.

Table.1. Parameter Settings of different biclustering methods

Biclustering Method	Parameter Settings
ISA	t_c = 2, seeds = 500, t_g = 2
SAMBA	N_1 = 4, D = 40, N_2 = 6, L = 30, k = 20
CC	$\delta \leq 0.5$ α = 1.2
Proposed FCM	S=3, K=25 N_bB_i=10.

The Table.2 shows non-overlapping modules with constant biclusters for increasing noise levels for the synthetic datasets. The Table.3 shows Overlapping modules with constant biclusters for increasing overlap degree for the synthetic datasets. The Table.4 shows non-overlapping modules with Additive Biclustering for increasing noise levels for the synthetic datasets. The Table.5 shows Overlapping modules with Additive Biclustering for increasing overlap degree for the synthetic datasets.

The results show that the proposed approach for modules without noise indicate over 85% biclustering than the ISA, SAMBA and CC method. The proposed method exceeds and retains a higher percentage for noise than other biclusters.

Table.2. Non-overlapping modules with constant biclusters for increasing noise levels for the synthetic Datasets

Overlap Degree	ISA	SAMBA	CC	Proposed FCM
0	0.3	0.3	0.1	0.44
0.05	0.24	0.25	0.1	0.52
0.1	0.28	0.275	0.1	0.51
0.15	0.28	0.275	0.15	0.5
0.2	0.29132	0.28	0.2	0.52
0.25	0.27	0.27	0.27	0.5

Table.3. Overlapping modules with constant biclusters for increasing overlap degree for the synthetic Datasets

Overlap Degree	ISA	SAMBA	CC	Proposed FCM
0	0.2	0.28	0.49	0.5
1	0.21	0.26	0.47	0.5
2	0.25	0.25	0.42	0.5
3	0.29	0.29	0.45	0.5
4	0.3	0.3	0.46	0.5
5	0.32	0.32	0.4	0.45
6	0.32	0.33	0.38	0.4
7	0.331	0.35	0.37	0.41
8	0.341	0.35	0.35	0.4

Table.4. Non-overlapping modules with Additive Biclustering for increasing noise levels for the synthetic Datasets

Overlap Degree	ISA	SAMBA	CC	Proposed FCM
0	0.24	0.21	0.175	0.31
0.05	0.31	0.22	0.15	0.32
0.1	0.28	0.23	0.15	0.33
0.15	0.28	0.275	0.15	0.3
0.2	0.28	0.29	0.15	0.313
0.25	0.27	0.31	0.16	0.33

Table.5. Overlapping modules with Additive Biclustering for increasing overlap degree for the synthetic Datasets

Overlap Degree	ISA	SAMBA	CC	Proposed FCM
0	0.23	0.29	0.13	0.49
1	0.3	0.27	0.12	0.51
2	0.3	0.25	0.13	0.5
3	0.3	0.26	0.14	0.495
4	0.32	0.23	0.13	0.49
5	0.4	0.22	0.15	0.42
6	0.35	0.25	0.14	0.45
7	0.3	0.26	0.15	0.43

8	0.3	0.2	0.14	0.41

5. CONCLUSION

In this study we proposed a large average submatrix-FCM bi-clustering algorithm for extracting gene expression data from microarray biclusters. The primary goal is to determine how highly correlated genes are optimally biclustered. Therefore, if the matrix lines and columns are instantaneously combined, the biclustering procedure is very effective. The first thing is to identify the set of sub matrices using the Large Average Submatrix. This is based on a simple sense ranking that transcends a series width and average value. A Large Average Submatrix is employed in an iterative manner, where there is a link between the highest value and the lowest description length. The matrix will increase and the clustering problem will be deficient with the overall increase in information from gene expression. At this point there are serious problems when information are enhanced by using the biclustering algorithm. Therefore, to increase biclustering, we use the Large Average Submatrix. This compresses or removes irrelevant or less correlated clustering results. The study uses FCM also to ensure that the number of rows and columns can be added to the sub-matrix for further calculation. The system is calculated for the accuracy of the components and the strength of the sub-matrices. The performance of the proposed method is provided with experiments on synthetic data sets. The experiments show that the proposed method is more competitive than any other biclustering algorithm with the given task. In future, the quality of biclustering may be increased through deep learning methods.

REFERENCES

[1] H. Bulut and A. Onan, "An Improved Ant-Based Algorithm Based on Heaps Merging and Fuzzy C-Means for Clustering Cancer Gene Expression Data", *Sadhana*, Vol. 45, No. 1, pp. 1-17, 2020.

[2] C. Lopez, S. Tucker and T., Salameh, "An Unsupervised Machine Learning Method for Discovering Patient Clusters based on Genetic Signatures", *Journal of Biomedical Informatics*, Vol. 85, pp. 30-39, 2018.

[3] S. Lee, "Fuzzy Clustering with Optimization for Collaborative Filtering-Based Recommender Systems", *Journal of Ambient Intelligence and Humanized Computing*, Vol. 52, 1-18, 2021.

[4] P. Edwin Dhas and B. Sankara Gomathi, "A Novel Clustering Algorithm by Clubbing GHFCM and GWO for Microarray Gene Data", *The Journal of Supercomputing*, Vol. 76, No. 8, pp. 5679-5693, 2020.

[5] I. Aljarah, M. Habib, H. Faris and S. Mirjalili, "Introduction to Evolutionary Data Clustering and Its Applications.", *Proceedings of International Conference on Evolutionary Data Clustering: Algorithms and Applications*, pp. 1-21, 2021.

[6] M. Fratello, L. Cattelani, A. Federico, and D. Greco, "Unsupervised Algorithms for Microarray Sample Stratification", *Proceedings of International Conference on Microarray Data Analysis*, pp. 121-146, 2022.

[7] D. Yan, H. Cao, Y. Yu and X. Yu, "Single-Objective/Multiobjective Cat Swarm Optimization Clustering Analysis for Data Partition", *IEEE Transactions on Automation Science and Engineering*, Vol. 17, No. 33, pp. 1633-1646, 2020.

[8] N. Kushwaha, M. Pant, S. Kant and V.K. Jain, "Magnetic Optimization Algorithm for Data Clustering", *Pattern Recognition Letters*, Vol. 115, pp. 59-65, 2018.

[9] Y. Yan and F.C. Harris, "A Survey of Data Clustering for Cancer Subtyping", *International Journal for Computers and Their Applications*, Vol. 28, No. 2, pp. 1-13, 2021.

[10] M. Franco and J.M. Vivo, "Cluster Analysis of Microarray Data", *Proceedings of International Conference on Microarray bioinformatics*, pp. 153-18, 2019.