Comparing Biclustering Algorithms: A Simulation Study

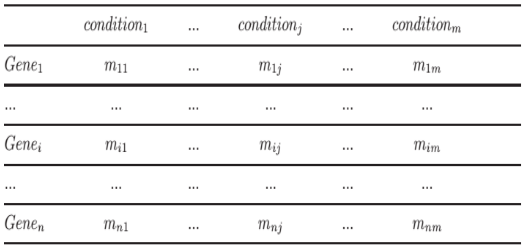
A. HOMAIDA, A. KOCATÜRK, B. ALTUNKAYNAK

Abstract— Biclustering is a data mining technique that simultaneously apply clustering to both of rows and columns of a matrix that represents a data set to detect local patterns in the data. Biclustering methods are developed originally to analyze gene expression data or any data that can be presented as a matrix. For this purpose, many biclustering algorithms have been introduced in the literature. However, efforts to compare the performance of these algorithms are limited. The purpose of this work is to compare the performance of biclustering algorithms. FLOC, Qubic, CTWC, SAMBA and Bimax algorithms were chosen in our work. The data contains four biclusters generated randomly with four different normal distributions. Algorithms showed varies performance with the simulated data type. To obtain the full information from the data more than one algorithm must be used. In addition, the chosen algorithm must be repeated many times with different parameters to have the best results.

Index Terms— Biclustering, Chia and Karuturi Function, Clustering, Gene Expression Data, Heatmaps.

1 INTRODUCTION

There are various types of data in our days. One of the most important data set is the gene expression dataset. In this dataset, the expression level of a number of genes is measured under some experimental conditions. The data will be arranged in a data matrix where each row represent one gene and each column is an experimental condition as follow[1]:



	$condition_1$	...	$condition_j$	...	$condition_m$
$Gene_1$	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

Many methods or techniques were introduced until now to analyze big data sets like gene expression data. The most used techniques are the data mining technique, which aims to discover patterns in large datasets. Data mining includes some methods at the intersection of machine learning, database systems and statistics.

Clustering method can be used with the gene expression data. However, clustering can be applied to either the rows or the columns separately. That means using clustering will produce clusters of rows or columns. In addition, in a given subject cluster, each subject is defined using all the variables. In clustering, the clusters are exhaustive. To overcome the above problems a newer method was introduced in 1972 by J. A. Hartigan [2], which allows clustering the rows and the columns in the same time. That means this method seeks blocks of rows and columns that are interrelated. In addition, the clusters, which are detected by this method, should not be exclusive or exhaustive. This method is known as biclustering method or block clustering[3]. The differences between the clustering and the

- *Ammar HOMAIDA Gazi University Department of Statistics Ankara, Turkey E-mail: ammar.homaida@gazi.edu.tr*
- *Ahmet KOCATÜRK Gazi University Department of Statistics Ankara, Turkey E-mail: ahmetkocaturk@gazi.edu.tr*
- *Bülent ALTUNKAYNAK Gazi University Department of Statistics, Turkey E-mail: bulenta@gazi.edu.tr*

biclustering methods are presented in the following figure:

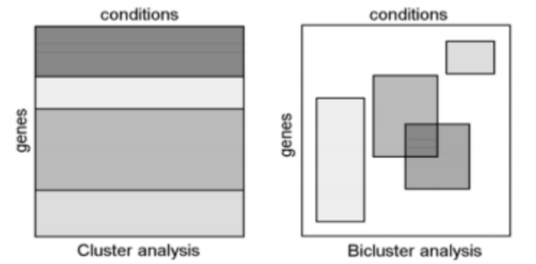


Fig. 2. The main difference between using clustering (left) and biclustering (right) methods.

1.1 Biclustering

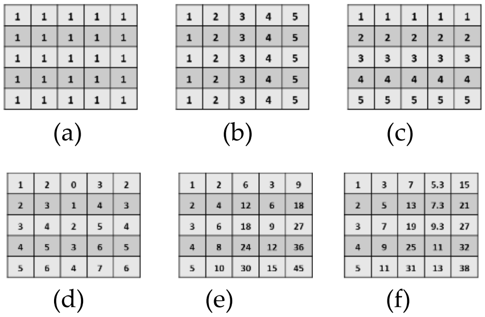
Notation: Let A be a data matrix with n rows and m columns. Let $X = \{x_1, ..., x_n\}$ be the rows of the A matrix, and $Y = \{y_1, ..., y_m\}$ be the columns of the matrix. The matrix A is detonated as (X, Y). Let $A_{IJ} = (I, J)$ where $I \subseteq X$ and $J \subseteq Y$ be a submatrix that contains only the elements a_{ij} corresponding to the rows I and columns J where $i \in I$ and $j \in J$. A bicluster is a submatrix $A_{IJ} = (I, J)$, where $I = \{i_1, ..., i_k\} \subseteq X$ and $J = \{j_1, ..., i_s\} \subseteq Y$ [4].

Many biclustering algorithms are introduced until now. Algorithms were classified according to the types of the biclusters, which can be detected in the data or the number, and the positions of the founded bicluster and some other classifications.

The first classification is made based on the type of the biclusters that can be detected in the data matrix. This classification is made by Madeira and Oliveira [5]. Biclustering algorithms can identify one or more of the following types:

1. Biclusters with constant values.
2. Biclusters with constant values on rows or columns.
3. Biclusters with coherent values.
4. Biclusters with coherent evolutions.

The previous four types are presented in the following Figures:



1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

(a)

1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5

(b)

1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5

(c)

1	2	0	3	2
2	3	1	4	3
3	4	2	5	4
4	5	3	6	5
5	6	4	7	6

(d)

1	2	6	3	9
2	4	12	6	18
3	6	18	9	27
4	8	24	12	36
5	10	30	15	45

(e)

1	3	7	5.3	15
2	5	13	7.3	21
3	7	19	9.3	27
4	9	25	11	32
5	11	31	13	38

(f)

Fig. 3. Examples of Different Types of Biclusters: (a) constant values (b) constant values on rows (c) constant values on columns (d) coherent values - additive (e) coherent values – multiplicative (f) coherent values – additive and multiplicative

The second classification is also made by Madeira and Oliveira [5]. The situation here whether there only just one bicluster in the data matrix (see Fig. 4 (a)) or more than one. In the second case, we have one of the following structure:

1. Exclusive row and column biclusters.
2. Non-Overlapping biclusters with checkerboard structure.
3. Exclusive-rows biclusters.
4. Exclusive-columns biclusters.
5. Non-Overlapping biclusters with tree structure.
6. Non-Overlapping non-exclusive biclusters.
7. Overlapping biclusters with hierarchical structure.
8. Arbitrarily positioned overlapping biclusters.

The previous structures are presented in the following figures:

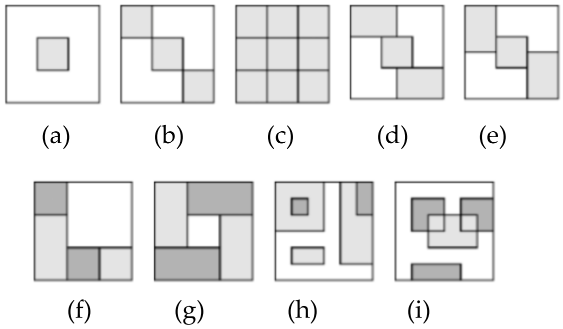


Fig. 4. Bicluster Structure: (a) Single Bicluster (b) Exclusive row and column biclusters (c) Non-Overlapping biclusters with checkerboard structure (d) Exclusive-rows biclusters (e) Exclusive-columns biclusters (f) Non-Overlapping biclusters with tree structure (g) Non-Overlapping non-exclusive biclusters (h) Overlapping biclusters with hierarchical structure (i) Arbitrarily positioned overlapping biclusters

3 RELATED WORK

There is a good number of studies that address the issue of comparing the biclustering algorithms. J. A. Hartigan introduced Block Clustering algorithm [2] which known as the first algorithm in the biclustering method. The block clustering algorithm was not applied to gene expression dataset. After that, Cheng and Church [6] introduced the CC algorithm in 2000 which known as first biclustering algorithm that used with gene expression data and opened the door for introducing many better algorithms.

One of the most important studies is the one which was done by Madeira and Oliveira [5]. That because in this work, they made the best classification for the founded biclustering algorithm. In addition, any new algorithm is classified using their work. They classified the biclustering algorithm according to the type, the number and the positions of the biclusters that can be discovered by a biclustering algorithm, and the methods used to perform the search.

Prelic et al. [7] provide in their paper a method for comparing the biclustering algorithms with introducing Bimax algorithm as a reference model. in addition, they showed that biclustering results have has advantages over a conventional hierarchical clustering approach.

Chia and Karuturi [8] proposed differential co-expression framework and a differential co-expression scoring function to objectively quantify quality or goodness of a bicluster of genes based on the observation that genes in a bicluster are co-expressed in the conditions belonged to the bicluster and not co-expressed in the other conditions. In addition, they proposed a scoring function help to classify the founded biclusters in three types: strong gene effects only, strong condition effect only, strong gene and condition effect).

Saber, H. B., and Elloumi, M., [9], made a new survey for the clustering and the biclustering algorithms and evaluation functions. Pontes et al.[10] also provide in their work a review of a large number of quality measures for gene expression biclusters.

4 SELECTED ALGORITHMS

4.1 FLOC Algorithm

The FLOC algorithm (Flexible Overlapping biClustering) presented by Yang et al. [11] works to detect k biclusters simultaneously with mean residues value are less than a user-defined threshold. The biclusters that are detected by this algorithm generally are arbitrarily positioned overlapped biclusters with coherent values. In addition, this algorithm follows the greedy iterative search strategy to find the biclusters in the data matrix.

FLOC algorithm based on adding rows or columns to a bicluster or deleting them if they were not originally in the bicluster. The process of adding or moving out rows or columns depends on the value of the gain score. The row or column with the best gain score will reduce the residual value for the bicluster. The process will be repeated again and again until no action can be done.

FLOC algorithm was introduced to overcome some of the problems of the CC algorithm [6]. In CC algorithm, when algorithm detects a bicluster it replaces it with random values, which may affect the data in the matric. FLOC algorithm does not replace the founded biclusters and can find k biclusters in the same time. FLOC algorithm depends on the initial biclusters which means it may fail to find a good result sometimes. In addition, detecting a big number of biclusters require long run time in most of the cases.

4.2 CTWC Algorithm

Getz et al. [12] introduced CTWC (Coupled Two-Way Clustering) algorithm to analyze the gene expression datasets. CTWC algorithm follows iterative row and column clustering combination strategy. That means CTWC algorithm applies a clustering method to rows and columns, separately. That will produce clusters of rows and columns. The next step is building the biclusters using some sort techniques. CTWC algorithm can be used using any known clustering algorithm.

CTWC algorithm uses only subsets of rows or columns. These subsets are identified, as stable clusters in previous clustering iteration are candidates for the next iteration. Getz et al. used a clustering algorithm called Super Para-magnetic Clustering (SPC) algorithm [13] in their work. SPC algorithm used to obtain stable clusters. This process stops when no new stable cluster can be founded.

CTWC algorithm works to detect arbitrarily positioned overlapped biclusters with constant column values. The data must be normalized before applying the CTWC algorithm.

4.4 SAMBA Algorithm

SAMBA algorithm (Statistical Algorithmic Method for Bicluster Analysis) which introduced by Tanay et al. [14], is a method that uses both of graph technique and statistical modeling to detect biclusters from the dataset which are statistically significant. In addition, SAMBA algorithm perform class discovery and feature selection, simultaneously.

SAMBA algorithm works to detect arbitrarily positioned overlapped biclusters with coherent evolution data type. In addition, SAMBA algorithm may search for biclusters with constant values. The main idea in SAMBA algorithm is to present the data as a bipartite graph. The nodes in the graph correspond to genes and conditions. In this graph, an edge between row (gene) and column (condition) represents the change in the expression value. This change may be up-regulation or down-regulation for a gene under different experimental conditions.

SAMBA algorithm follows exhaustive bicluster enumeration strategy. This strategy works on the idea that to find the best biclusters, an exhaustive search is being applied for all of the possible biclusters in the dataset. SAMBA algorithm may not deal well with datasets that have a high level of noise.

4.3 Bimax Algorithm

Prelić et al. [7] introduced a fast divide-and-conquer strategy based algorithm called Bimax algorithm. In divide and conquer strategy, instead of working directly with the whole problem, algorithm breaks it into several subproblems. That means the difference between the original problem and the sub-problems just the size. By solving these sub-problems, we can obtain the solution for the original problem.

Bimax Algorithms works to detect biclusters with checkerboard structure and coherent evolution values type. This algorithm is used as a reference algorithm in many works. Bimax algorithm can work with the logical data type. Therefore, before applying Bimax algorithm the data must be converted into a binary matrix using a user-predefined threshold. Algorithm has three main steps: first, the row and columns will be arranged so the ones values are in the upper right corner of the matrix. The second step is to divide the matrix into two part. Finally, we return the matrix that has just ones in it as a bicluster. Then, repeat the process until we have all of the possible biclusters.

4.5 Qubic Algorithm

Li et al. [15] introduced qualitative biclustering (Qubic) algorithm for analyses the gene expression data sets. Qubic algorithm works to detect non-exhaustive or non-exclusive biclusters with coherent values using distribution parameter identification strategy. Algorithms that use this strategy assume the data structure has a statistical model. Then it tries to fit the parameters to the data by minimizing a certain criterion through an iterative approach.

Qubic algorithm can deal with big datasets better from many other biclustering algorithms. The detected biclusters using Qubic algorithm may also be overlapped or not using a control parameter for that. In addition, it can detect both positively and negatively correlated expression patterns. Another advantage of using Qubic algorithm, it can deal better with many algorithms with the data that have outliers in it. Qubic algorithm is able to find many types of the biclusters especially constant columns or rows bicluster types with a good run time.

5 DATA and COMPARISON

The data matrix, which will be used in this work to compare the selected algorithms, was artificially generated. The matrix contains 300 rows and 250 columns. The background data were generated using the standard normal destitution. Then four non-overlapped biclusters with four different distributions were generated in the matrix, as we can see in the heatmap figure:

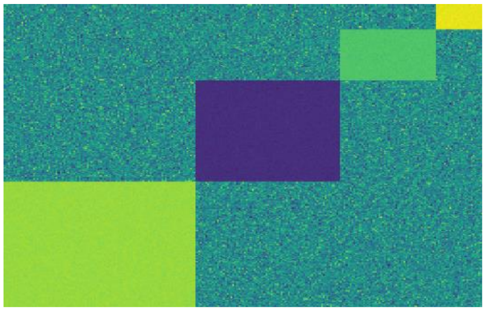


Fig. 5. Heatmap of the test data

The comparison is based on the whole founded size and the differential co-expression score SB(b)[8] where b is a bicluster. The strong positive score indicates strong co-expression in G_1 (the set of the conditions that are included in the founded bicluster) and weaker or no co-expression in G_2 (the set of the conditions that are not included in the founded bicluster) vice versa. (for more detail see [8]).

In this work, we are interested in the biggest SB score and size.

4.6 Results

The 5 algorithms were used with the simulated data matrix, and the results are including the names of the used packages:

Table 1: Table of the results

Algorithm	package	Size	SBscore(average)
FLOC	BicARE[16]	82.5%	5.819449
CTWC	CTWC[17]	74.67%	5.526198
Bimax	biclust[18]	74%	5.526198
Qubic	QUBIC[19]	37.05%	3.835679
SAMBA	Expander[20]	11.232%	2.829093

As we can see from the table, FLOC algorithm has the best results. It was able to detect about 82.5% of the whole size with the highest SB score (which was taken as the average value for the founded bicluster). The results for FLOC algorithm are presented in the following heatmap figure:

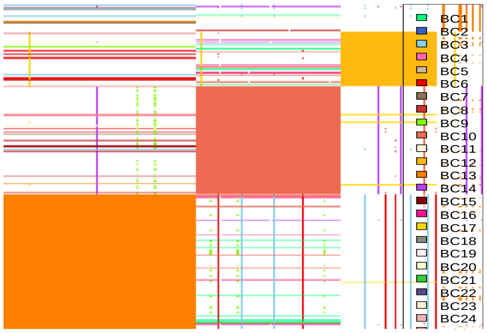


Fig. 6. Heatmap of the FLOC results

CTWC algorithm also showed good results, which was able to detect about 75% of the whole size with high, positively SB score as presented in the figure:

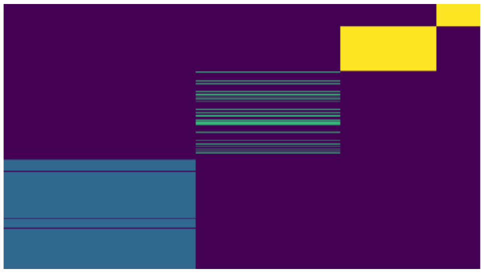


Fig. 7. Heatmap of the CTWC results

Bimax algorithm nearly has the same scores of the CTWC's results but with small differences between them as we can see in the figure:

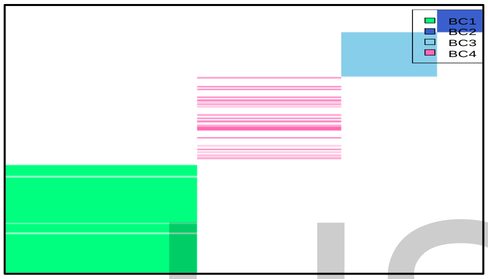


Fig. 8. Heatmap of the Bimax results

Qubic algorithm did not show a good performance with this dataset. It was able to detect about 37% of the whole size, as we can see in the figure:

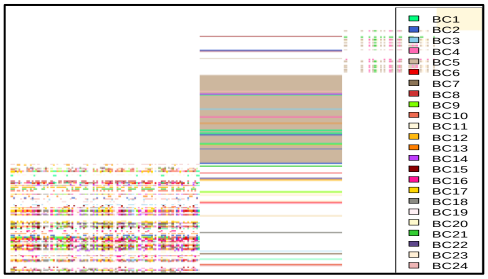


Fig. 9. Heatmap of the Qubic results

Finally, with SAMBA algorithm we were not able to have important results with this dataset. It was able to detect about 11% of the whole size as we can see in the heatmap figure:

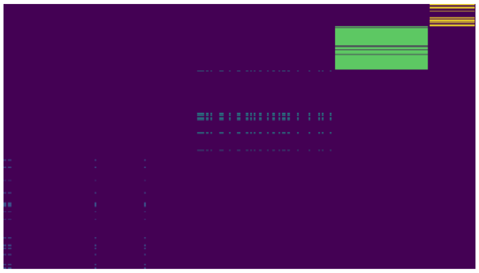


Fig. 10. Fig. 6. Heatmap of the SMABA results

6 DISCUSSION

Biclustering is a powerful method that helps us to detect submatrices with specific patterns. In addition, it can be used in any field if the data can be presented as a data matrix. However, as we have seen from the results and the previous works we cannot depend completely on one algorithm to obtain the full information from the data. Another problem is to choose the right parameters for every algorithm which not an easy job. So to have the best results we must choose more than one algorithm and every algorithm should be run with different parameters according to the data type.

REFERENCES

[1] Freitas, A., Ayadi, W., Elloumi, M., Oliveira, L.J., Hao, J.K, *A Survey on Biclustersing of Gene Expression Data.* Biological Knowledge Discovery Handbook: Preprocessing, Mining, and Postprocessing of Biological Data, 2013: p. 591-608.

[2] Hartigan, J.A., *Direct Clustering of a Data Matrix.* Journal of the American Statistical Association, 1972. **67, No. 337**: p. 123-129.

[3] Vittapu, M.S., *Synchronized Clustering: A Review on Systematic Comparisons and Validation of Prominent Block-Clustering Algorithms.* International Journal of Engineering and Information Systems (IJEAIS), 2017. **1**(4): p. 28-37.

[4] Leite, C.A.M., *Domain Oriented Biclustering Validation*, in *Department of Computer Science*. 2016, University of Porto.

[5] Madeira, S.C., & Oliveira, A.L., *Biclustering Algorithms for Biological Data Analysis: A Survey.* IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2004. **1**: p. 24-45.

[6] Y. Cheng, G.M.C., *Biclustering of Expression Data.* Int. Conf. Intelligent Systems for Molecular Biology, 2000. **12**: p. 61–86.

[7] Prelic, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W., Hennig, L., Thiele, L., & Zitzler, E., *A systematic comparison and evaluation of biclustering methods for gene expression data.* Bioinformatics, 2006. **22**(9): p. 1122-9.

[8] Chia, B., & Murthy, K.R., *Differential Co-Expression Framework to Quantify Goodness of Biclusters and Compare Biclustering Algorithms.* Algorithms for Molecular Biology, 2010: p. 5-23.

[9] Saber, H.B., & Elloumi, M., *A New Survey on Biclustering Tools, Biclustering Validation and Evaluations.* 2015.

[10] Pontes, B., Girldez, R., & Aguilar-Ruiz, J.S., *Quality Measures for Gene Expression Biclusters.* PLoS ONE 2015. **10**(3): p.: e0115497. doi:10.1371/journal. pone.0115497.

[11] J. Yang, H.W., W. Wang, and P.S. Yu,, *Enhanced Biclustering on Expression Data.* Proc. Third IEEE Conf. Bioinformatics and Bioeng, 2003. **3**: p. 321-327.

[12] Getz, G., Erel Levine, and Eytan Domany, *Coupled Two-Way Clustering of DNA Microarray Data.* Proceedings of the National Academy of Sciences 97.22, 2000: p. 12079-12084.

[13] Blatt M., W.S., Domany E., *Superparamagnetic Clustering of Data.* PHYSICAL REVIEW LETTERS, 1996. **76**(18): p. 3251-3254.

[14] Tanay, A., Sharan, R., & Shamir, R. , *Discovering Statistically Significant Biclusters in Gene Expression Data.* Bioinformatics, 2002. **18**(suppl_1): p. S136–S144.

[15] Li, G., Ma, Q., Tang, H., Paterson, A.H., & Xu, Y. , *Qubic: a qualitative biclustering algorithm for analyses of gene expression data.* Nucleic acids research, 2009. **37**(15): p. e101.

[16] Gestraud P., B.I., Barillot E., *BicARE : Biclustering Analysis and Results Exploration.* http://bioinfo.curie.fr, 2017.

[17] Getz G., D.E., *Coupled Two-Way Clustering Server.* BIOINFORMATICS 2003. **19**(9): p. 1153–1154.

[18] Kaiser S., S.R., Khamiakova T., Sil M., Theron R., Quintales L., Leisch F., Troyer E., *Package 'biclust'.* R Package "https://cran.r-project.org/web/packages/biclust/index.html", 2015.

[19] Zhang Y., M.Q., *Package 'QUBIC'.* R package "http://bioconductor.org/packages/release/bioc/html/QUBIC.html", 2018.

[20] Sharan R, M.-K.A., Shamir R, *EXPANDER: a system for clustering and visualizing gene expression data.* Bioinformatics, 2003. **19**(14): p. 1787–1799.