A Binary Biclustering Algorithm based on the Adjacency Difference Matrix for Gene Expression Data Analysis

HeMing Chu
Qufu Normal University
Xiangzhen Kong (kongxzhen@163.com)
Qufu Normal University
Jinxing Liu
Qufu Normal University
Juan Wang
Qufu Normal University
Chunhou Zheng
Qufu Normal University
Ke Zhang
People's Hospital of Rizhao

Research Article

Keywords: biclustering, gene expression data, adjacency matrix, binary data

Posted Date: May 27th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1633057/v1

A Binary Biclustering Algorithm based on the Adjacency Difference Matrix for Gene Expression Data Analysis

He-Ming Chu, Jin-Xing Liu, Ke Zhang, Chun-Hou Zheng, Juan Wang, Xiang-Zhen Kong

***Abstract*—Biclustering algorithms are very effective tools for processing gene expression datasets. Biclustering methods can be divided into two main categories, which are binary biclustering and non-binary biclustering. A binary matrix is usually converted from pre-processed gene expression data, which can effectively reduce the interference from noise and abnormal data, and is then processed using a binary biclustering algorithm. In this paper, we propose a new binary biclustering algorithm to deal with binary matrices, called the Adjacency Difference Matrix Binary Biclustering algorithm (AMBB) to address the drawback that most binary biclustering algorithms could not efficiently handle large gene expression data. AMBB constructs the adjacency matrix based on the adjacency difference values, and the output clusters are submatrices obtained by continuously updating the adjacency matrix that is constructed from the adjacency differences matrix. The adjacency matrix allows for clustering of genes that undergo similar reactions under different conditions into clusters, which is important for subsequent genes analysis. Meanwhile, experiments on synthetic and real datasets visually demonstrate that the AMBB algorithm has high practicability.**

***Index Terms*—biclustering, gene expression data, adjacency matrix, binary data**

I. Introduction

In recent years, the binary data matrix has been used in a variety of fields, including bioinformatics [1, 2], data mining [3-5], data analysis [6], etc. A binary dataset is a data matrix about the relationship between a set of objects[7]. The only two elements in the binary matrix are 0 and 1. For the binary biclustering algorithm that processes the binary data matrix converted from gene expression data, the resulting biclustered clusters are considered to be submatrices of all 1. And a bicluster is considered statistically significant when the number of 1's in it is significant enough [8]. Noteworthy, the meaning of 0 and 1 in the binary matrix could be known by combining the context. In this paper, the values 1 and 0 represent that the gene reacts or not under certain conditions, respectively. Up to now, a host of binary biclustering algorithms have been investigated by researchers. For example, the Bimax algorithm proposed by Prelić *et al.* [9], which is a iteration algorithm that could get maximal binary submatrix, where the important information represents 1, otherwise 0. The BiBit algorithm proposed by Domingo *et al. [7]* is to obtain biclusters by row coding. In the binary matrix, the element values have the same meaning as Bimax. Furthermore, the BiBinAlter algorithm proposed by Saber *et al.* [10], the Binary Matrix Factorization (BMF) proposed by Zhang *et al.* [11], the QUBIC2 algorithm proposed by Xie *et al.* [12] and so on are excellent binary biclustering algorithms. Specially, the QUBIC2 algorithm use various preprocessing methods to convert gene expression data into a binary data matrix and obtain biclustering. Noteworthy, the columns with value 1 in the binary matrix represent the same features. Biclustering methods are NP-hard [13]. Therefore, according to the solution of the algorithm, the biclustering algorithms can be classified into various types [14, 15]. Nevertheless, two major categories can be classified from the data, binary biclustering and non-binary biclustering algorithms. Comparing the binary biclustering algorithm with the non-binary biclustering algorithm, it can be found that two kinds of biclustering algorithms deal with different datasets. The non-binary biclustering algorithms can process gene expression data directly, such as Local Search Algorithms (LSM) [16], while binary biclustering algorithms convert gene expression data into a binary data matrix before processing the expression dataset. For the binary biclustering algorithm, one of the termination conditions of the Bimax algorithm is when the resulting submatrix no longer contains 0 elements. In addition, the BiBit algorithm eventually obtains biclustered clusters also by selecting columns with a value of 1 in the encoding. The non-binary biclustering algorithm will obtain the biclustered clusters directly according to its algorithmic steps. Cheng and Church are the first to apply the biclustering algorithm to gene expression data and this algorithm is named Cheng and Church (CC) [17]. It uses the mean squared residue value (MSR) method to add elements by putting each element into the seed and calculating the MSR value against a set threshold. When the MSR value of an added element is greater

H.-M. Chu, J.X, Liu, J. Wang are with School of Computer Science, Qufu Normal University, Rizhao 276826, China (e-mail: chuheming@yeah.net; sdcavell@126.com; wangjuansdu@163.com;).

K. Zhang (Corresponding author) is with Department of Oncology, Rizhao People's Hospital, Rizhao, 276826, China ((e-mail: zhangke96888@163.com).

C.-H. Zheng is with the School of Information Science and Engineering, Qufu Normal University, Rizhao 276826, China (e-mail: zhengch99@126.com).

X.-Z. Kong (Corresponding author) is with Computer Science, Qufu Normal University, Rizhao 276826, China (e-mail: kongxzhen@163.com).

The GO enrichment analysis website is available online at https://david.ncifcrf.gov/tools.jsp.

than the threshold value, it means that the element cannot be added to the seed set. The Scaling mean squared (SMSR) method, which is similar to the CC algorithm, is proposed by Mukhopadhyay *et al*. [18]. This algorithm is also based on MSR values to obtain biclustered. Direct clustering (DC) proposed by Hartigan *et al.* [19] is among the first published biclustering algorithms applied to data matrices. The Flexible Overlapped biclustering (FLOC) algorithm proposed by Yang *et al.* [20] is a stochastic iterative greedy algorithm. This algorithm is to initialize k biclustered and adds rows and columns to the biclusters according to the given probability.

The binary biclustering algorithm belongs to the special biclustering algorithm, which mainly processes the binary matrix in order to obtain the optimal biclusters. In this paper, a new binary biclustering algorithm based on constructing adjacency difference matrices is proposed, called the Adjacency Difference Matrix Binary Biclustering (AMBB) algorithm. The AMBB algorithm solves the problem of clustering genes with similar responses under certain conditions into groups. Based on the adjacency difference matrix, AMBB can effectively cluster genes that are closely related under certain conditions. In addition, the performance of the AMBB algorithm is tested with synthetic and real datasets. The experimental results show that the AMBB algorithm outperforms the BiBit and the Bimax in the synthetic, and the AMBB algorithm can obtain a large number of valid genes in the real dataset, which is very important for analyzing the gene expression data further.

II. METHODS

The BiBit algorithm uses the parameters $\min_r$ and $\min_c$ to obtain the final bicluster. And at least the $\min_r$ and $\min_c$ both equal 2. However, AMBB algorithm without such limitation and therefore, it can accurately identify each data item and test the similar genes under the current conditions. In addition, the AMBB algorithm does not require encoding and traversing all rows to continuously obtain the seeds. It uses the row with the highest number of 1's in the binary matrix as the seed, and iterates the row and column elements continuously according to the adjacency difference matrix to obtain the bicluster. Significantly, the adjacency difference matrix is constructed based on the seed.

The parameters that need to be input for the algorithm are row adjacency difference matrix threshold δ and column adjacency difference matrix threshold λ, where the δ is used to control the selection of rows and the λ is used to control the selection of columns. Details are described in section II.

A. Declaration

An input pre-processed binary data matrix is defined as $E=(I,J)$, where I and J are two finite sets, denoting the set of rows and the set of columns, respectively. For gene expression data, we define rows to represent genes and columns to represent conditions. Furthermore, gene $i \in I$ reacts to under condition $j \in J$ when the value of element x_{ij} in the binary matrix is 1, otherwise 0.

The binary data matrix $E=(I,J)$, with $n=|I|$ and $m=|J|$, can be constructed as a $n*n$ row difference matrix and a $m*m$ column difference matrix. The elements of the rows perform $\wedge(AND)$ operations against the elements of the seed and the values obtained are summed up, and this value is called the row difference value $\eta_{ii'}$. Where the i' represents the i-th row. The column difference value $\eta_{jj'}$ is calculated in the same way. And as same as the row difference value, the j' represents the j-th column.

In the row difference value matrix, each row represents the vector of value differences between the seed and all rows in the binary matrix. Similarly, the rows and columns of the column difference value matrix are expressed in terms of the same meaning as the row difference value matrix.

The submatrix E' is called the maximum biclustering cluster if there are no elements with value 0 in this submatrix

B. Parameter

In the AMBB algorithm, there are two parameters that are the row difference threshold δ and the column difference threshold λ. These two threshold functions are used to select relevant rows and relevant columns. The row difference value matrix was constructed based on i-th ($1 \le i \le n$) row that the i'-th row when the $\eta_{ii'}$ is smaller than the row threshold δ is clustered to form a row set. And then the column difference value matrix was constructed based on j-th ($1 \le j \le m$) column that the j'-th column when the $\eta_{jj'}$ is smaller than the column threshold λ is clustered to form a sub-matrix. Therefore, it is very important to choose a suitable threshold value. When an algorithm is able to get the best threshold automatically, then it will be easier to get the best cluster for the operation of this algorithm. In this paper, we propose a new method for semi-automatic selection of threshold values. A range is manually given according to the size of the dataset, and the optimal threshold is recorded by a circular method, and the optimal bicluster is obtained at the same time. This approach solves the challenge of selecting the best threshold in the AMBB algorithm.

Considering that the sizes of the binary data matrix are not the same, the density of 1 is also different. A threshold range is given to AMBB, and the algorithm is run once for each threshold, keeping the optimal clustering obtained by this algorithm. When the AMBB algorithm runs all the thresholds, the one of them best is selected according to the number of clusters obtained. The threshold is determined by the number of biclusters obtained, and a higher number indicates a higher threshold. To get the similar columns, the threshold of columns is initially set to 3. Each iteration of the column threshold in the same submatrix will automatically subtract one until the value is 0. The selection of row thresholds is illustrated in Figure 1. Noteworthily, the 1's density in synthetic dataset is set 50%. We use three different sizes of synthetic data to select the threshold with the maximum number of biclusters, where each dataset is run 100 times.

We consider the threshold that the maximum number of t biclusters obtained is most likely to be the optimal row

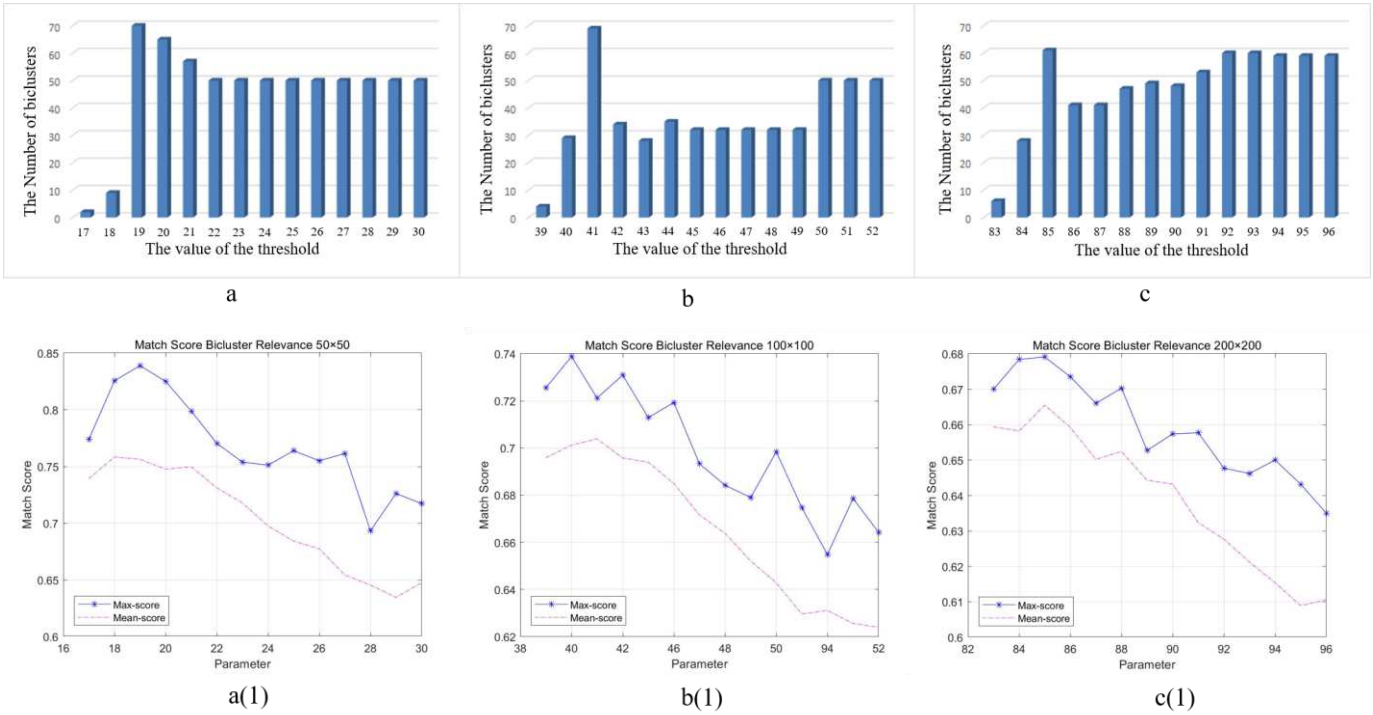


Figure 1. A schematic of the biclustered clusters obtained for each row threshold. (a) The number of biclusters result of the synthetic dataset of 50*50. (b) The number of biclusters result of 100*100 synthetic dataset. (c) The number of biclusters result of 200*200 synthetic dataset.

threshold, and use the threshold as a benchmark to find the optimal threshold.

In Figure 1, the a (1) shows the match score of thresholds in 50*50 synthetic dataset and the optimal row threshold is 19. The b (1) shows the match score of thresholds in 100*100 synthetic dataset and the optimal row threshold is 40. In 200*200 synthetic dataset, the c (1) shows the match score and the optimal row threshold is 85.

Noteworthily, running ten times for each dataset, the Max-score represents maximum score and the Mean-score represents mean score. To summarize the selection of row thresholds, we only need to input a threshold range, and the AMBB algorithm finds the threshold within the input range that yields the maximum number of clusters, and then uses this threshold as a benchmark to find the optimal parameter for the current data.

C. *Algorithm*

The schematic diagram of the algorithm is shown in Figure 2. The main steps of the algorithm are the selection of the seeds and the construction of the adjacency difference matrix. We have two ways to obtain some seeds. The first way is to use each row as a seed and construct a row difference matrix based on each row. The other is to use the row with the highest number of 1's in it as a seed first. Figure 3 shows two methods in detail. In order to find the optimal method, we compared these two approaches, and detailed results are presented in the next section.

Fig. 2(a) represents the input binary matrix. Fig. 2(b) shows the selection of seeds, which is simply the selection of the row with the highest number of 1 from the binary matrix. Five rectangles exist in Fig. 2(c), representing the five difference matrices obtained from the five seeds, each of which contains a row difference matrix and a column difference matrix. Fig. 2(d) is the final output of three biclustered clusters, as seeds may not have biclustered clusters according to the threshold value.

Once the seeds are selected, the AMBB algorithm constructs a row difference matrix and a column difference matrix for the seed. Figure 3 depicts the two obtained seed methods of AMBB algorithm, denoted as method a (a-AMBB) and method b (b-AMBB). There are two ways to construct the difference matrix by the two methods. Method b selects the row with the largest row value as the seed, and then constructs as row difference matrix based on that row, where the matrix dimension is $1*m$. The difference value (DV) is defined as follows:

$$DV_{si} = \sum_{j=1}^{m} (x_{ij} \wedge x_{sj}), \tag{1}$$

where DV_{si} denotes the difference value between seed s and row i. The x_{ij} denotes the j-th column of the i-th row, and the x_{sj} denotes the j-th column of the seed.

If the DV value is less than the threshold δ, the i-th row is put into the row cluster. When all the rows have been computed, a cluster of rows expanded by seeds is then obtained. The seed in Figure 3 is the fourth row of the binary matrix, and the difference value matrix is $1*10$. Note that in this example, we set a row threshold δ of 5. Put rows with difference value less than 5 together to form a row cluster. The column values are calculated for each column in the row cluster, and the column values are calculated as follows:

$$CV_j = \sum_{i \in I', j \in J} x_{ij}, \tag{2}$$

where CV_j denotes the column value of the j-th column. The I' indicates row cluster. The column with the largest

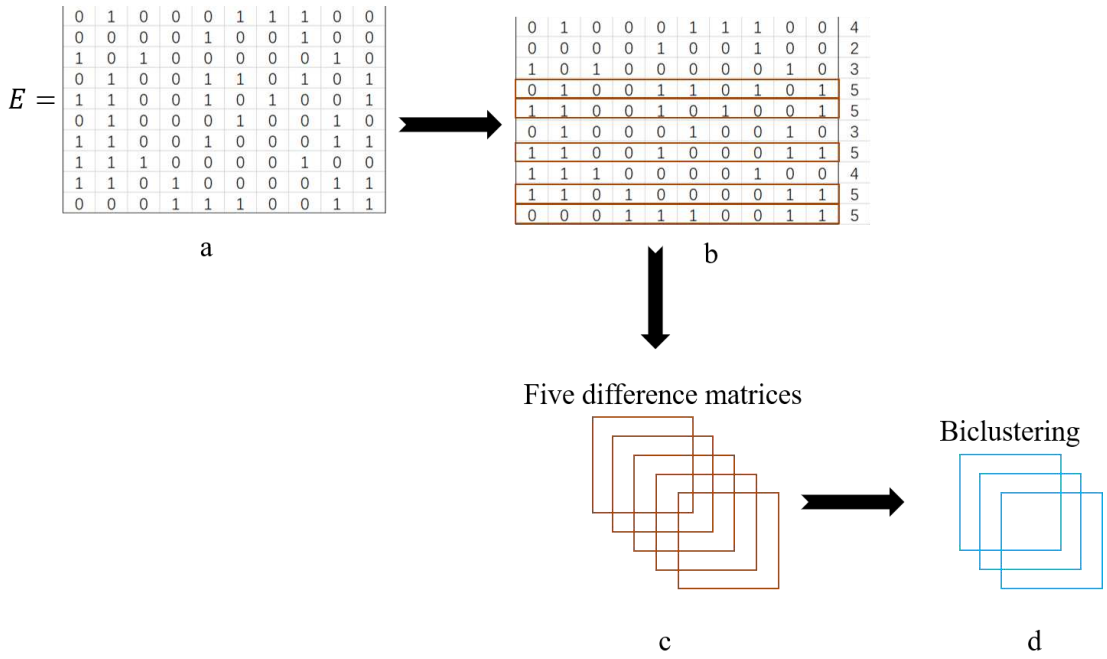


Fig. 2. A brief schematic of the AMBB algorithm. (a) shows the pre-processing matrix. (b) shows the selection of seeds. (c) shows the construction of the difference matrix, and Figure 2(d) shows the acquisition of biclusters.

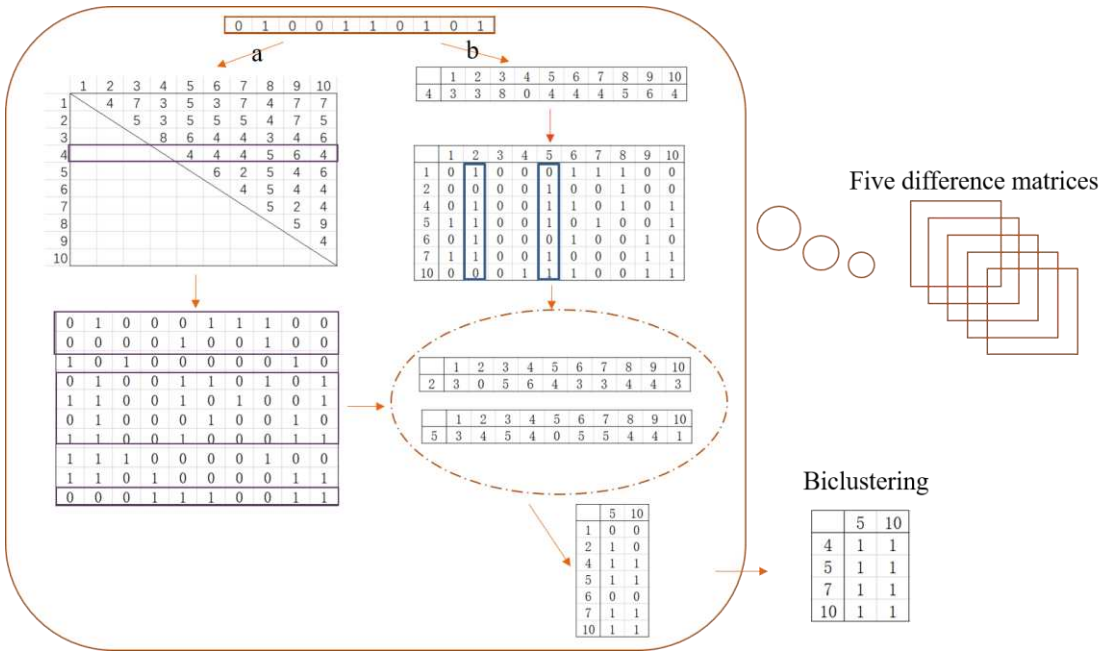


Figure 3. The diagram of two methods for two obtained seed methods of AMBB. Methods a and method b show two methods of obtaining seeds.

column value in I' is given priority as the column seed, and the column difference matrix is constructed. Normally the column threshold λ is set to 1 for better acquisition of clusters. In the example of Figure 3, the second column does not have a matching column, so no submatrix can be obtained. However, the fifth column and the tenth column could construct a column cluster. When the obtained submatrix contains element 0, repeat the above steps for that submatrix until all the elements are 1.

Method a differs from method b in that method a uses all rows as seeds once to obtain clusters. Thus method a is able to obtain more biclustered clusters. Through experimental comparison, the performance of these two methods does not differ significantly, but the operation speed of method b is much lower than that of method a, and detailed comparison results will be given in the next section. In this thesis, Table 1 shows the step code of method b.

III. Results

In this section, the performance of the AMBB algorithm is evaluated from two aspects. The first part illustrates AMBB method and from the synthetic dataset and selects the best method from the two methods to compare with the Bimax and BiBit algorithms. For the synthetic dataset, the performance of

Table 1. AMBB biclustering algorithm

Input: E : Binary matrix
δ : row threshold
λ : column threshold
Output: X : final biclusters

1. for every seed of max row value x_i do
2. $\eta_{ii'} = \sum_{j \in J} x_{ij} \wedge x_{i'j}$
3. if $\eta_{ii'} \le \delta$ do
4. for every sub-seed of max column value x_j do
5. $\eta_{jj'} = \sum_{i \in I'} x_{ij} \wedge x_{ij'}$
6. if $\eta_{jj'} \le \lambda$ do
7. Add $x_{i'j'}$ to X
8. end if
9. end for
10. end if
11. end for

the AMBB algorithm is analyzed according to the different densities in the synthetic binary matrix. In the second part, the real datasets are used to illustrate the usefulness of the AMBB algorithm using GO enrichment analysis.

A. *Evaluation Metric*

For the binary biclustering algorithm, the commonly used evaluation metric is the match score. The details of the method are described in [9]. The Match Score is defined as follows:

$$S(E_1, E_2) = \frac{1}{|E_1|} \sum_{(G_1, C_1) \in E_1} \max_{(G_2, C_2) \in E_2} \frac{|G_1 \text{ I } G_2|}{|G_1 \text{ U } G_2|}, \quad (3)$$

where E_1 and E_2 are two sets of biclusters. G_1 is the row (gene) set of E_1. C_1 is the column (condition) set of E_1. Match score reflects the average of the maximum match scores between E_1 and E_2. When the value of S is 1, this means that the biclustered E_1 is consistent with the biclustered E_2.

B. *Synthetic Dataset*

Synthetic datasets are used to test the performance of two types of AMBB. In addition, the results obtained by these two methods of two AMBB are compared with the Bimax and BiBit algorithms. The synthetic dataset is divided into three sizes, namely 50*50, 100*100 and 200*200, where the density of 1's in these three synthetic datasets ranges from 5% to 50%, increasing by 5% each time. In contrast to the experiments with synthetic datasets in the BiBit algorithm, in this experiment, the density of 1's is randomly distributed, in order to simulate the uncertainty of the dataset.

Since the binary synthetic datasets are all similar, we experiment with the optimal parameters set by the BiBit algorithm, $\min_r$ =2 and $\min_c$ =2. Through extensive experiments with the Bimax algorithm and ended up with the parameters $\min_r$ =2 and $\min_c$ =2 for the 50*50 and 100*100 datasets and 200*200 data sets $\min_r$ =5, $\min_c$ =5. Both of these algorithms are implemented in R.

First, Figure 4 (a), Figure 4 (b) and Figure 4 (c) show the performance of two ways methods of AMBB algorithm in synthetic dataset. Although the difference in match score is not larger, the scores of b-AMBB method are higher than a-AMBB method at low density. Furthermore, Figure 5 shows the running times of the four methods. The size of the synthetic dataset used is 100*100. In Figure 5, the vertical coordinate indicates the time taken to run the algorithm, and the horizontal coordinate indicates the density of 1's in the dataset. The two algorithms used by AMBB consume more time than the Bimax algorithm. At the same time, we can see from this that the time of b-AMBB is shorter than the time of the a-AMBB algorithm. Therefore, the AMBB algorithm in this thesis mainly refers to b-AMBB method.

Next, we compare the performance of the three methods. The detailed results are shown in Figure 4 (d) to Figure 4 (f). For 50*50 datasets, although the results are not as good as Bimax at low density, the AMBB algorithm has the best performance overall. The AMBB algorithm and the BiBit algorithm have similar trend, but the match scores of AMBB is higher than that of BiBit algorithm. In the 100*100 and 200*200 datasets, it is known that the Bimax algorithm has the lowest results, while the AMBB and the BiBit have about similar performance.

After analyzing the match scores of the three biclustering algorithms in the three synthetic datasets, it can be concluded that the AMBB algorithm is able to obtain the shortest amount of time and has the best performance when compared with other algorithms.

C. *Real Dataset*

To study the utility of the AMBB algorithm, a human gene expression dataset known as Pollen [21]and a mouse gene expression dataset, namely Buettner [22] are analyzed. In accordance with the preprocessing method of the Bimax algorithm [9], the dataset will first be normalized before the biclustering algorithm can word on it. The purpose of our analysis is to find biclusters in the dataset and to investigate the biological relevance of these genes. In the dataset, there are 14805 genes and 249 conditions. In addition, Table 2 exhibits the detailed information of the above two datasets. For those datasets, the original data is processed into a binary matrix using the same preprocessing method as the Bimax algorithm. To analyze the real dataset, we use GO enrichment analysis [23]. This is because genes in biclustered cells are involved in biological processes. Many methods are now available to implement GO enrichment analysis. We choose to use the David website because this website updated the database some years ago and has very rich features and simplicity of operation.

Table 2. The information of datasets

Dataset	Number of Gene	Number of condition	Number of type	Specie
Pollen	14805	80	11	Human
Buettner	8989	182	3	Mouse

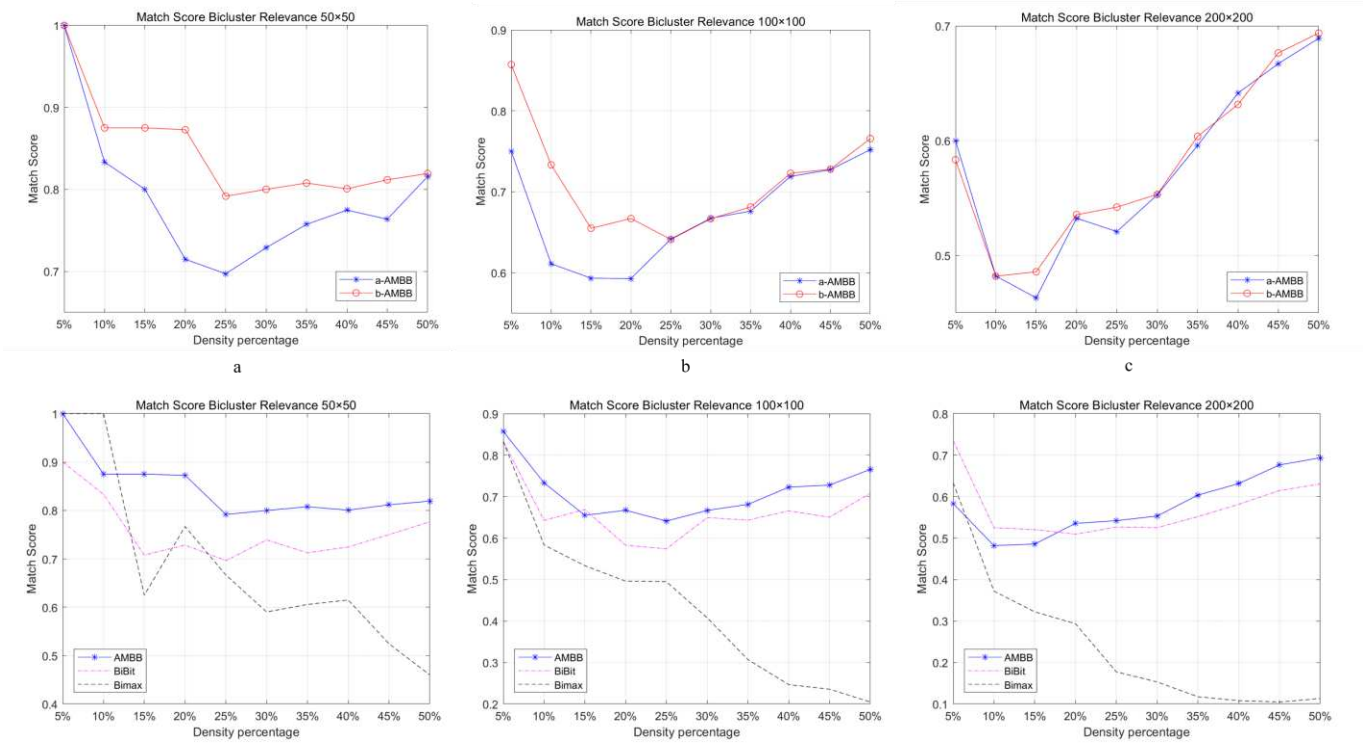


Fig. 4. Comparison of methods for synthetic datasets. Figure 4(a) to Figure 4(c) show the comparison of two AMBB methods. Figure 4(d) to Figure 4(f) show the comparison with four biclustering algorithms.

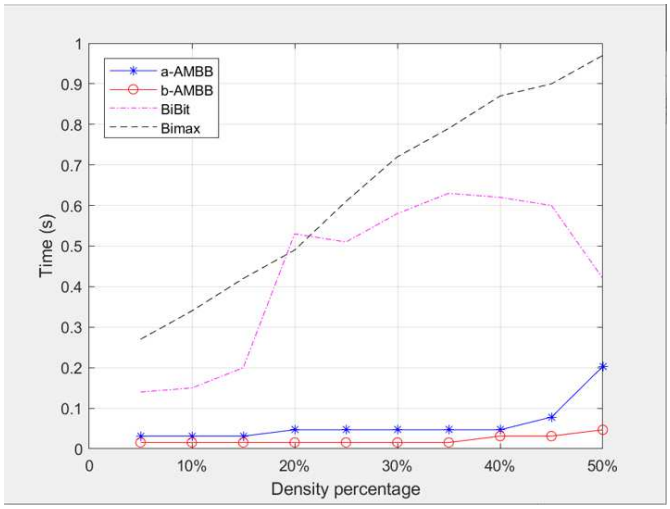


Figure 5. Time comparisons of the four methods are available.

In this experiment, we compared three algorithms, the AMBB algorithm, QUBIC algorithm, and Bimax algorithm, respectively. The QUBIC algorithm is run by the software in [24], and the Bimax algorithm is used the package bicluster in R version 4.1.2. The detailed step is proposed by S. Kaiser [25]. It is worth noting that the running time of the b-AMBB method is shorter than that the a -AMBB method. In the real data we default the AMBB algorithm to the b-AMBB method. Valid genes are used as the evaluation criteria, and genes are considered valid when the p-value is less than 0.05 [26]. The p-value is calculated as a hypergeometric distribution with the following formula:

$$p-value=\sum_{o=n}^{N}\frac{\binom{O}{o}\binom{T-O}{t-o}}{\binom{T}{t}}. \tag{4}$$

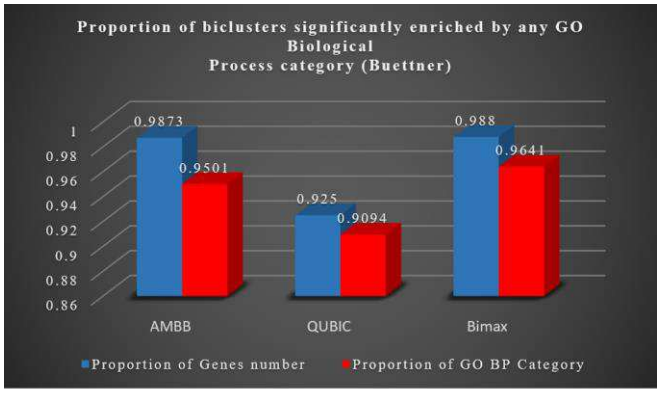
In this formula, O denotes the acquired genes and T denotes the total genes in the database.

Figure 6 shows the results of the effective genes obtained by the three algorithms. According to Figure 6, there exist two comparison results for Proportion of Genes Number and Proportion of BP Category. The two metrics are calculated as shown below:

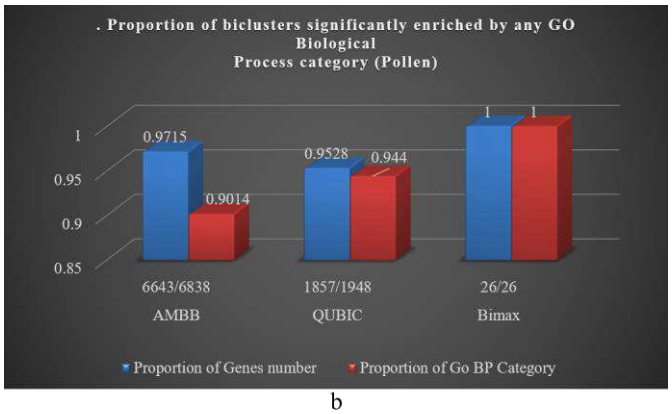
$$proportion_{GN}=\frac{R_G}{T_G}, \tag{5}$$

$$proportion_{GObp}=\frac{V_G}{T_G}. \tag{6}$$

Where the $proportion_{GN}$ represents the Proportion of Genes Number, and the R_G represents the number of identified GO items that the p-value is less than 0.05, and the T_G represents the total number of GO items. The $proportion_{GObp}$ and the V_G represent the Proportion of Biological Process (BP) Category and the number of highly enriched GO items in BP, respectively. The gene count ratio indicates the proportion of the identified genes out of the total genes obtained. Higher values indicate more genes in the biclustered that can be identified and the greater the utility of the algorithm. To analyze the genes obtained from the biclusters, we used David [27]. For the Figure 6(a), it is the result of Buettner dataset. The total number of genes in the biclustered obtained by the three algorithms are 6448, 773 and 502, the number of genes



a



b

Fig. 6. Comparison chart of effective genes. Figure 6(a) shows the proportion of valid genes in Buettner dataset. Figure 6(b) shows the proportion of valid genes in Pollen dataset.

identified are 6366, 715, 496, respectively, and the valid genes are 6126, 703 and 484. According to the Eq.5 and Eq. 6, the AMBB algorithm obtained the highest number of valid genes in the Buettner dataset, as well as a very high accuracy rate. By analysis the Figure 6(b), the total number of genes obtained by the AMBB algorithm is much higher than the other two methods, which also leads to the denominator number of the ratio being too large, but also reaching 0.9. This also shows that the utility of the AMBB algorithm has a high practicality. Although the Proportion of Genes Number value of AMBB is smaller than that of Bimax, the valid genes obtained by the AMBB algorithm are much higher than those obtained by the QUBIC and Bimax algorithm. Proportion of GO BP Category indicates the ratio of genes with p-value less than 0.05 to the genes in the obtained biclustered. Among these three methods, the AMBB algorithm has the lowest value.

IV. Conclusion

In this paper, we propose a new binary biclustering algorithm for gene expression data. It uses the construction of a neighbor-joining difference matrix to obtain similar genes. This approach has better time complexity than compared algorithms, and also has high practical, which will be useful for subsequent analyses. Although the AMBB algorithm has two thresholds, the row difference threshold and the column difference threshold, at runtime for the row threshold we only give a range, and the algorithm will automatically select the optimal clusters. In addition, the column threshold is fixed 3. For each iteration of column clustering, the column threshold is automatically reduced by one until the value is 0. After analyzing the comparison of the synthetic and real datasets, the performance of our method has been also visually demonstrated. However, the AMBB algorithm also has the disadvantage that the genes obtained are not as ideal as expected for the most efficient gene analysis. Owing to the fact that the binary matrix can only show valid and non-valid genes and cannot identify the importance of valid genes. Therefore, in the future study, we will try to fuse the weights to cluster similar genes. Theoretically, this maybe produce satisfactory results.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

Conceived and designed the experiments: KZ and HMC. Performed the experiment: HMC. Analyzed the data XZK and JXL. Contributed reagents/materials/analysis tools: CHZ. Contributed to the writing of the manuscript: XZK and HMC.

Ethics approval and consent to participate

Not applicable.

Consent to Publish

Not applicable.

Availability of data and materials

The dataset accession number by Pollen submitted at National Center for Biotechnology Information (NCBI) (http://www.ncbi.nlm.nih.gov/Traces/sra/) is SRP041736. In addition, the dataset accession number by Buettner submitted at ArrayExpresss (http://www.ebi.ac.uk/arrayexpress/) is E-MTAB-2805.

Funding

This work is funded by the grants of the National Science Foundation of China, Nos. 61702299 and jointly in part by National Natural Science Foundation of China, Nos. 61872220, 62172253.

Acknowledgements

Not applicable.

References

1. Zhang, Z., et al., *Binary multi-view clustering.* IEEE transactions on pattern analysis and machine intelligence, 2018. **41**(7): p. 1774-1782.
2. Eren, K., et al., *A comparative analysis of biclustering algorithms for gene expression data.* Brief Bioinform, 2013. **14**(3): p. 279-92.
3. Ayub, U. and S.A. Moqurrab. *Predicting crop diseases using data mining approaches: classification*. in *2018 1st International Conference*

On Power, Energy And Smart Grid (Icpesg). 2018. IEEE.

4. Alqadah, F., R. Bhatnagar, and A. Jegga, *A novel framework for detecting maximally banded matrices in binary data.* Statistical Analysis and Data Mining: The ASA Data Science Journal, 2010. **3**(6): p. 431-445.
5. Colantonio, A., et al. *ABBA: Adaptive bicluster-based approach to impute missing values in binary matrices*. in *Proceedings of the 2010 ACM Symposium on Applied Computing*. 2010.
6. Negrín-Hernández, M.-A., M. Martel-Escobar, and F.-J. Vázquez-Polo, *Bayesian meta-analysis for binary data and prior distribution on models.* International Journal of Environmental Research and Public Health, 2021. **18**(2): p. 809.
7. Rodriguez-Baena, D.S., A.J. Perez-Pulido, and J.S. Aguilar-Ruiz, *A biclustering algorithm for extracting bit-patterns from binary datasets.* Bioinformatics, 2011. **27**(19): p. 2738-45.
8. Koyuturk, M., W. Szpankowski, and A. Grama. *Biclustering gene-feature matrices for statistically significant dense patterns*. in *Proceedings. 2004 IEEE Computational Systems Bioinformatics Conference, 2004. CSB 2004.* 2004. IEEE.
9. Prelic, A., et al., *A systematic comparison and evaluation of biclustering methods for gene expression data.* Bioinformatics, 2006. **22**(9): p. 1122-9.
10. Saber, H.B. and M. Elloumi, *A novel biclustering algorithm of binary microarray data: BiBinCons and BiBinAlter.* BioData Min, 2015. **8**: p. 38.
11. Zhang, Z.-Y., et al., *Binary matrix factorization for analyzing gene expression data.* Data Mining and Knowledge Discovery, 2010. **20**(1): p. 28-52.
12. Xie, J., et al., *QUBIC2: a novel and robust biclustering algorithm for analyses and interpretation of large-scale RNA-Seq data.* Bioinformatics, 2020. **36**(4): p. 1143-1149.
13. Cheng, K.-O., et al., *Identification of coherent patterns in gene expression data using an efficient biclustering algorithm and parallel coordinate visualization.* BMC bioinformatics, 2008. **9**(1): p. 1-28.
14. Saber, H.B. and M. Elloumi, *A comparative study of clustering and biclustering of microarray data.* International Journal of Computer Science & Information Technology, 2014. **6**(6): p. 93.
15. Pontes, B., R. Giráldez, and J.S. Aguilar-Ruiz, *Biclustering on expression data: A review.* Journal of biomedical informatics, 2015. **57**: p. 163-180.
16. Maâtouk, O., et al., *Local search method based on biological knowledge for the biclustering of gene expression data.* Advances in Smart Systems Research, 2012. **6**(2): p. 65.
17. Cheng, Y. and G.M. Church. *Biclustering of expression data*. in *Ismb*. 2000.
18. Mukhopadhyay, A., U. Maulik, and S. Bandyopadhyay, *A novel coherence measure for discovering scaling biclusters from gene expression data.* Journal of bioinformatics and computational biology, 2009. **7**(05): p. 853-868.
19. Hartigan, J.A., *Direct clustering of a data matrix.* Journal of the american statistical association, 1972. **67**(337): p. 123-129.
20. Yang, J., et al., *An improved biclustering method for analyzing gene expression profiles.* International Journal on Artificial Intelligence Tools, 2005. **14**(05): p. 771-789.
21. Pollen, A.A., et al., *Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex.* Nature biotechnology, 2014. **32**(10): p. 1053-1058.
22. Buettner, F., et al., *Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells.* Nat Biotechnol, 2015. **33**(2): p. 155-60.
23. Consortium, G.O., *The gene ontology (GO) project in 2006.* Nucleic acids research, 2006. **34**(suppl_1): p. D322-D326.
24. Verma, N.K., et al., *BIDEAL: A Toolbox for Bicluster Analysis—Generation, Visualization and Validation.* SN Computer Science, 2021. **2**(1): p. 1-15.
25. Kaiser, S. and F. Leisch, *A toolbox for bicluster analysis in R.* Compstat 2008 – Proceedings in Computational Statistics, 2008. **HeidelbergPhysica Verlag(pg. 201-208)**.
26. Orzechowski, P., et al., *EBIC: an evolutionary-based parallel biclustering algorithm for pattern discovery.* Bioinformatics, 2018. **34**(21): p. 3719-3726.
27. Huang, D.W., et al., *DAVID Bioinformatics Resources: expanded annotation database and novel algorithms to better extract biology from large gene lists.* Nucleic acids research, 2007. **35**(suppl_2): p. W169-W175.