An Improved Biclustering Algorithm for Gene Expression Data

Sheng-Hua Jin* and Li Hua

School of Computer Engineering, Huaiyin Institute of Technology, Huaian, Jiangsu, 223003;

Huaian key Laboratory of the Study and Application of Internet of Things, Huaian, Jiangsu, 223003;

Jiangsu "Internet of Things" Mobile Internet Technology Engineering laboratory, Huaian, Jiangsu, 223003

Abstract: Cheng-Church (CC) biclustering algorithm is the popular algorithm for the gene expression data mining at present. Only find one biclustering can be found at one time and the biclustering that overlap each other can hardly be found when using this algorithm. This article puts forward a modified algorithm for the gene expression data mining that uses the middle biclustering result to conduct the randomization process, digging up more eligible biclustering data. It also raised a parallel computing method that uses the multi-core processor or cluster environment to improve efficiency. It is proved by experimental verification that the modified algorithm enhances the precision and efficiency of the gene expression data mining to a certain degree.

Keywords: Biclustering, gene expression, parallel mining, randomization.

1. INTRODUCTION

With the accomplishment of the human genome project in 2003, the scientists began to switch their focus on the study of the gene functional groups, hoping to use the gene expression [1] data to reveal the genes' internal functions and learn their mechanism of interaction, thus reveal the mystery of human life and improve the quality of people's lives. In order to dig up useful biological information from the vast amounts of gene expression data, scientists have put forward many different kinds of clustering algorithms, such as the traditional clustering algorithm raised by GETZ G [2], QU [3] etc., which divides the genes into disjoint categories in a certain condition and the genes in one category are similar.

But the similarity degrees of the genes under different conditions are different; therefore we can only seek the overall information instead of the partial information that contains vast connotative biological information. In 2000 Cheng & Church (CC) first put forward the concept of (Bicluster) [4] that the clustering can be conducted on multiple rows and columns at the same time in a data matrix, enabling the clustering of gene expression data in two dimensions by the gene and the experiment condition, therefore we can find which genes under which conditions have similar expression levels or close relations. But Cheng & Church (CC) biclustering [5-7] can only conduct serial computation, and some of the biclustering results can be meaningless. This article puts forward a modified algorithm of Medium Cheng-Church (MCC) which uses the middle biclustering result to conduct the randomization [8] process and reduce the influence of randomization on the biclustering results. It not only increases the amount of eligible biclustering results but also solves the present problem that the biclustering can't do parallel computing, improving computation efficiency greatly.

2. CC BICLUSTERING ALGORITHM

Set X as gene set, and Y as the corresponding expression conditions set. aij are elements in the gene expression data matrix M. Set I, J as the subset of X, Y respectively, then (I, J) has the following average square residues to the specified sub-matrix:

$$a_{i,J}= \frac{1}{|J|}\sum_{j \in J} a_{ij} \tag{1}$$

$$a_{Ij}= \frac{1}{|I|}\sum_{i \in I} a_{ij} \tag{2}$$

$$a_{I,J}= \frac{1}{|I||J|}\sum_{i\in I,j\in J} a_{ij} \tag{3}$$

$$RS_{I,J}(i,\ \ j)= \frac{1}{|I||J|}\sum_{i\in I,j\in J}\left(a_{ij}\text{-}a_{i,J}\text{-}a_{Ij}\text{-}a_{I,J}\right)^2 \tag{4}$$

$$H(I,J) = \sum_{i \in I\ ,j \in J}\frac{RS_{I,J}^2}{|I||J|} \tag{5}$$

Among which, I is the original matrix column value, J is the original matrix row value. For the row average value of matrix M, the column average value of matrix M, the average value of all elements in matrix M and the average square residue of matrix, CC biclustering adopts the thought of greedy algorithm and find a biggest possible result of biclustering sub-matrix. And the average square residue of the biclustering sub-matrix result is less than the certain threshold value given. Define the tetrad Cheng-Church: of which U is the condition, V is the gene data, E is the gene expression matrix and δ is the square residue. The algorithm steps of CC biclustering are:

*Address correspondence to this author at the Huaiyin Institute of Technology, Huaian, 223003, China; Tel: +13852319363;
E-mail: 13852319363@139.com

1874-110X/14 2014 Bentham Open

Evaluate the scores of every column and row in the gene expression matrix E, pick the biggest row i, the biggest column j, compare the magnitude of $d(i)$ and $e(j)$, remove the i row or j column of the higher score in the gene expression matrix.

$$d(i) = \frac{1}{|j|}\sum_{j\in J'} RS_{I,J}(i,j) \tag{6}$$

$$e(j) = \frac{1}{|i|}\sum_{i\in I'} RS_{I,J}(i,j) \tag{7}$$

Average square residue statistic: Compute the average square residue of the rest gene expression matrix, if it is larger than the threshold value, repeat the second step, if it's not, then obtain the preliminary biclustering results and go on to the next step.

According to formula (6) and (7), compute the scores of every column and row, pick row i with the biggest d(i), and column j with the biggest e(j), compare the magnitude of d (i) and e(j).

Combine the i row or the j column of the smaller score value with the biclustering result M and compute the average square residue, if it's smaller, then add the i row or the j column of the smaller score value into the biclustering result and repeat the forth step; if it's bigger, then M is the final biclustering result.

In the actual situation, one operation of CC biclustering can only work out one biclustering result. The literature 3 points out that in order to discover more biclustering results, Chen and church suggest the application of repeated modified matrix to the biclustering algorithm. The modification is the randomization value on the previously found biclustering result, thus to avoid that the related signals go against the finding of other biclustering results in the gene expression matrix. CC biclustering has the following features: 1, only one biclustering result in the original matrix can be obtained in one iterative calculation; 2, Next iterative calculation can only be conducted after the randomization of the original matrix based on the biclustering results of last calculation. According to the second feature, the CC biclustering results can only conduct serial computation, and because of the randomization process of the original matrix, some biclustering results cannot be worked out, and some of the biclustering results can be meaningless.

3. THE INPROVED METHODS OF MCC

In one computation we can work out vast middle biclustering results, and these results all meet the condition of being greater than the average square residues. Thus we can conduct the randomization process many times in one iterative calculation process, and in this way the original matrix can be turned into many different matrix, can we can conduct biclustering computation on different matrix and conduct duplicate removal to the computed biclustering results. Thus we can not only obtain more eligible biclustering results but also avoid the shortcoming that the CC biclustering can only conduct serial computation, using parallel computing and reducing the computation time. The detailed algorithm is as the following:

```
Input: lined up matrix M, $a_{i,j} \in M_c$, in which $a_{i,j} \in M_c, 0 < i \le n, 0 < j \le n, \delta, m$
Output: biclustering set $M_c$
1 define $a_{i,J}= \frac{1}{|J|}\sum_{j\in J} a_{ij}$
2 define $a_{Ij}= \frac{1}{|I|}\sum_{i\in I} a_{ij}$
3 define $a_{I,J}= \frac{1}{|I||J|}\sum_{i\in I,j\in J} a_{ij}$
4 define $RS_{I,J}(i,j) = \frac{1}{|I||J|}\sum_{i\in I,j\in J}(a_{ij}-a_{i,J}-a_{Ij}-a_{I,J})^2$
5 define $H(I,J) = \sum_{i\in I,j\in J}\frac{RS^2_{I,J}}{|I||J|}$
6 Initialize a bicluster $\{I',J'\}$ with $\{I' = I, J' = J\}$
7 Delete phase
8 While $(H(I',J') > \delta)$ do
9 Compute for $i \in I', d(i) = \frac{1}{|j|}\sum_{j\in J'} RS_{I,J}(i,j)$
10 Compute for $j \in J', e(j) = \frac{1}{|i|}\sum_{i\in I'} RS_{I,J}(i,j)$
11 If $\max_{i\in I} d(i) > \max_{j\in J} e(j)$ $assign\ I = I \setminus \{argmax_i(d(i))\}$
12 Else $J = J \setminus \{argmax_i(e(j))\}$
13 End While
14 If $\max(\mathrm{Size}(I'), \mathrm{Size}(J')) > m$
15 Assign $M' = random(M)$
16 Goto 6
End if
17 Addition phase
18 Assign $I'' = I', J'' = J'$
19 While $(H(I'',J'') < \delta)$ do
20 Compute for $i \in I/I', d(i) = \frac{1}{|j|}\sum_{j\in J''} RS_{I,J}(i,j)$
21 Compute for $j \in J/J', e(j) = \frac{1}{|i|}\sum_{i\in I''} RS_{I,J}(i,j)$
22 If $\max_{i\in I} d(i) > \max_{j\in J} e(j)$ Assign $I'' = I'' \cup argmax(e(j))$
23 Else $J'' = J'' \cup argmax(d(i))$
24 End if
25 End While
26 If $\max(\mathrm{Size}(I''), \mathrm{Size}(J'')) > m$
27 Assign $M' = random(M)$
28 Goto 6
29 End if
30 If $M_c \notin \{I'',J''\}$
31 = $M_c \cup \{I'',J''\}$
```

4. EXPERIMENT RESULT ANALYSIS

4.1. Experimental Data Set Introduction

The data used in the experiment come from the gene expression database AGEMAP [9-10]. AGEMAP records the change of gene expression data along with the change of ages of mice. It includes 8932 genes and 16896 cDNA from 16 tissues, including 5 male and female mice of age 1 month, 6 months, 16 months, and 24 months respectively. By analyzing AGEMAP we can find those age-related genes. This article chooses the muscle data set of it and conduct experimental verification using MCC algorithm steps, and the threshold values are set respectively 0.02, 0.025, 0.03, 0.035 in the experiment process.

4.2. Precision Comparison with CC Algorithm

Compare the efficiency of MCC algorithm and CC algorithm. Both algorithms dig up those related biggest biclusterings of gene expression values from the original gene expression data. They both adopt muscle data set. Fig. (**1**) gives the clustering results of two algorithms.

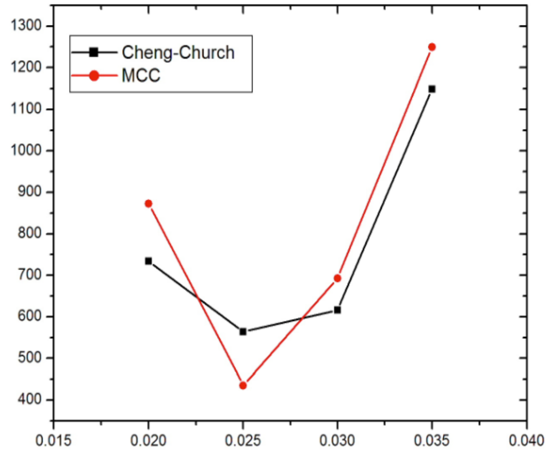

Fig. (1). Comparison of the experiment results of MCC algorithm and CC algorithm.

In Fig. (**1**), the horizontal axis shows the chosen threshold values, and the vertical axis are the eligible biclustering numbers computed by the algorithm. Form Fig. (**1**) we can see that in the threshold is about sections of about less than 0.020 and more than 0.030, the biclustering numbers computed by MCC is more than that of CC, that’s mainly because MCC follows the thought of “plus after minus” of the CC algorithm and first conduct minus operation to the original matrix and obtain the biclustering result of the smallest magnitude, and then conduct plus operation to the biclustering result of the smallest magnitude, thus obtain the optimized biclustering solution. Meanwhile because the randomization process of MCC biclustering algorithm doesn’t have to wait for the completion of its last iterative operation, but is directly operated on the original matrix after the completion of cutting nodes. This can obtain more biclustering results by reducing the modification of the original matrix.

4.3. Comparison of Computation Efficiency

In order to verify the computation efficiency of MCC, the processing time of MCC and CC on the muscle data set is compared. MCC computes more matrix than CC, and MCC needs to save a large amount of middle matrix in the computation process, while the internal storage capacity is far from enough and extra expense will be cost. Therefore, in the one-machine situation, the computation efficiency of MCC is lower than that of CC. But MCC can conduct parallel computation, therefore this experiment verifies whether the computation of MCC in the parallel computation situation is higher than the computation efficiency of CC. The experiment platform is Intel 4 core processor 3.4GHZ, with 8G internal storage, and 64-bit linux operating system. Simulate the parallel computation on the one machine with mapreduce [11, 12], and because the machine is 4-cored, we simulate the parallel computation of 4 machines. Fig. (**2**) shows the computation time of the two algorithms on the data set.

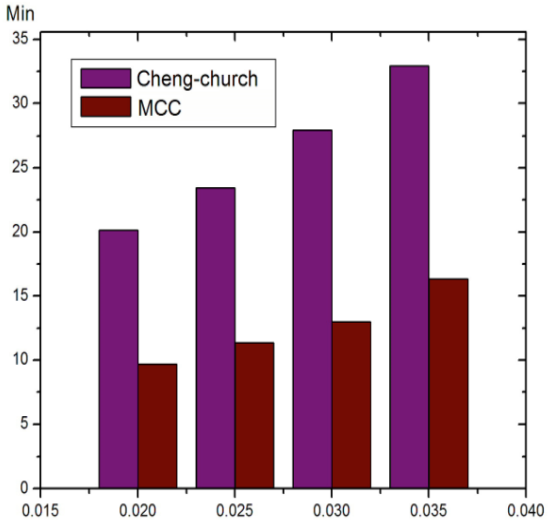

Fig. (2). The computation time of MCC algorithm and CC algorithm.

From Fig. (**2**), we can see that the computation time of MCC algorithm is almost half of that of the computation time of CC algorithm, which improves computation efficiency greatly. If multiple computers conduct parallel computation at the same time, it will enhance the computation efficiency to a larger degree.

CONCLUSION

The biclustering method allows the clustering of gene expression data on two dimensions of the gene and the experiment condition. This article puts forward a modified clustering algorithm MCC which accelerates the original CC computing speed and obtains the comparatively more accurate precision. It also raises a parallel computing method that uses the multi-core processor or cluster environment to improve efficiency. Compared to the previous CC algorithm, the parallel computing is more applicable to large-scale data analysis.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This work is supported by the National natural fund project, China (No. 61402192), Jiangsu natural science fund project, China (No. 14KJB520006).

REFERENCES

[1] O. Voggenreiter, S. Bleuler, W. Gruissem, “Exact biclustering algorithm for the analysis of large gene expression data sets”, *BMC Bioinformatics*, vol. 13(Suppl 18), pp. A10, 2012.

[2] G. Getz, E. Levine, and E. Domany, “Coupled two-way clustering analysis of gene microarray data”. In Proceedings of the Natural Academy of Sciences USA, pp. 12079-12084, 2000.

[3] J. B. Qu, N. Michael, L. N Chen, “Constrained subspace clustering for time series gene expression data”. In 4th International Conference on Computational Systems Biology, pp. 9-11, 2010.

[4] K. Bryan, M. Terrile, I. M. Bray, “Discovery and visualization of miRNA–mRNA functional modules within integrated data using bicluster analysis”. *Nucleic Acids Research*, vol. 42, pp. e17-e17, 2014.

[5] A. Serin, “Biclustering analysis for large scale data”, Freie Universität Berlin: Germany, 2012.

[6] Y. Cheng and G. M. Church. “Biclustering of expression data”, Proc. Eighth Int’l Conf. Intelligent Systems for Molecular Biology (ISMB’00), pp. 93-103, 2000.

[7] T. Saito and Y. Okada, “Bicluster-Network Method and Its Application to Movie Recommendation”. *Knowledge and Systems Engineering*. pp. 147-153, 2014.

[8] J. A. Spertus, D. J. Maron, and D. J. Cohen, “Frequency, predictors, and consequences of crossing over to revascularization within 12 months of randomization to optimal medical therapy in the Clinical Outcomes Utilizing Revascularization and Aggressive Drug Evaluation (COURAGE) trial”. *Circulation: Cardiovascular Quality and Outcomes*, vol. 6, pp. 409-418, 2013.

[9] J. M. ZAHN and P. S. AGEMAP, “A Gene Expression Database for Aging in Mice”, *PLOS Genetics*, vol. 3, pp. 2326-2337, 2007.

[11] Y. Sun, N. R. Zhang, A. B. Owen, “Multiple hypothesis testing adjusted for latent variables, with an application to the Agemap gene expression data”. *The Annals of Applied Statistics*, vol. 6, pp. 1664-1688, 2012.

[12] J. MacQueen, “Some Methods for Classification and Analysis of MultiVariateObservations”, Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability. pp. 281-297, 1967.

Received: September 16, 2014 Revised: December 23, 2014 Accepted: December 31, 2014