An Improved Biclustering Algorithm and Its Application to Gene Expression Spectrum Analysis

Hua Qu[1,2], Liu-Pu Wang[3], Yan-Chun Liang[3]*, and Chun-Guo Wu[3,4]

[1] *College of Software, Key Laboratory of Symbol Computation and Knowledge Engineering of the Ministry of Education, Jilin University, Changchun 130012, China;* [2] *Guangzhou Institute of China Telecom, Guangzhou 510630, China;* [3] *College of Computer Science and Technology, Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China;* [4] *Key Laboratory of Information Science and Engineering of the Ministry of Railway/Key Laboratory of Advanced Information Science and Network Technology of Beijing, Beijing Jiaotong University, Beijing 100044, China.*

Cheng and Church algorithm is an important approach in biclustering algorithms. In this paper, the process of the extended space in the second stage of Cheng and Church algorithm is improved and the selections of two important parameters are discussed. The results of the improved algorithm used in the gene expression spectrum analysis show that, compared with Cheng and Church algorithm, the quality of clustering results is enhanced obviously, the mining expression models are better, and the data possess a strong consistency with fluctuation on the condition while the computational time does not increase significantly.

Key words: biclustering algorithm, gene expression pedigree analysis, Cheng and Church algorithm

Introduction

Gene expression spectrum analysis is an important subject in the field of bioinformatics. Its task is to find remarkable structures from the data matrix. The structure types consist of the overall and the local models. As an effective tool to analyze gene expression data, the clustering analysis is comprehensively applied to many fields, such as gene expression spectrum analysis (*1*), genome study, biological regulatory networks (*2*), medicine filtering, new medicine development, clinical disease diagnosis (*3*, *4*), and so on. The basic hypothesis included in the clustering analysis is that, the genes that have the same expression mode may have similar functions (*5*). However, traditional clustering methods have a series of problems in reducing noise, mining local information, and synthesizing the heterogeneous data, *etc.*

The data dimension is becoming higher and higher due to the use of new biological microarray chips. Different objects of the same cluster in the data of a high dimensional space could show the similarity only in a certain subspace. When this principle is applied to gene expression data, mutual-controlling genes could show similar expression patterns in some conditions for test samples. In fact, in the whole input space, the gene expression pattern is different. Therefore, difficulties appear when we use traditional clustering methods to determine the object similarity by using the value approximation in high dimensional data.

The biclustering algorithm presented by Cheng and Church (*6*) is different from traditional clustering algorithms, in which the similarity is not treated as a function of pairs of genes or pairs of conditions. Instead, it is a measure of the coherence of the genes and conditions in the biclustering. This measure can be a symmetric function of genes and conditions involved and thus the finding of biclusters is a process that groups genes and conditions simultaneously.

Cheng and Church algorithm

The most important innovation of Cheng and Church algorithm is that they put forward a definition called *residue score* (*6*, *7*). The algorithm divides an expression model into three parts: attribute residue, object residue, and δ-cluster residue (or background residue). The mathematical definitions are as follows:

$$e_{Ij} = \frac{\sum_{i \in I} e_{ij}}{|I|} \qquad e_{iJ} = \frac{\sum_{j \in J} e_{ij}}{|J|}$$

*** Corresponding author.**
E-mail: ycliang@jlu.edu.cn

$$e_{IJ} = \frac{\sum_{i\in I, j\in J} e_{ij}}{|I||J|} \quad (1)$$

where I and J are the row and column vector sets of the submatrix, respectively; $|I|$ and $|J|$ are the number of rows and columns, respectively; e_{ij} is the element of the submatrix; e_{Ij}, e_{iJ}, and e_{IJ} are the attribute residue, object residue, and δ-cluster residue, respectively.

The definition of the residue score is as follows:

$$RS_{IJ}(i,j) = e_{ij} - e_{Ij} - e_{iJ} + e_{IJ} \quad (2)$$

Let X be the set of genes and Y the set of conditions. Let e_{ij} be the element of the gene-condition expression matrix representing the logarithm of the relative abundance of the mRNA of the i^{th} gene under the j^{th} condition. Let $I \subset X$ and $J \subset Y$ be the subsets of genes and conditions. The pair (I, J) specifies a submatrix A_{IJ} with the following *mean squared residue score*:

$$H(I,J) = \sum_{i\in I, j\in J} \frac{RS_{ij}^2}{|I||J|} \quad (3)$$

The lowest score $H(I, J) = 0$ indicates that the gene expression levels fluctuate in unison. This includes the trivial or constant biclusters where there is no fluctuation. These trivial biclusters may not be very interesting but need to be discovered and masked so that more interesting ones can be found. The row variance may be an accompanying score to reject trivial biclusters:

$$V(I,J) = \frac{1}{|J|}\sum_{j\in J}(e_{ij} - e_{Ij})^2 \quad (4)$$

The higher the value of H is, the more disordered the data is. In Cheng and Church algorithm, a greedy method is used to select submatrix with a low H score. It is divided into two phases. Firstly, the method is to remove the row or column to achieve the largest decrease of the score. For the current submatrix, they calculate the average residue score of each row using $d(i) = \frac{1}{|J|}\sum_{j\in J} RS_{IJ}(i,j)$ and the average residue score of each column using $e(j) = \frac{1}{|I|}\sum_{i\in I} RS_{IJ}(i,j)$, then choose the row or column with the maximal score and delete it from the current submatrix, until $H(I, J) < \delta$. Also they use a parameter α, so that they can delete a set of nodes each time before the score is recalculated. Without updating the score after the removal of each node, the matrix may shrink too much and one may miss some large δ-clusters. One may also choose an adaptive α based on the score and the size during the iteration. Secondly, they add rows and columns so that the matrix with the maximal size can be obtained.

Results and Discussion

Improvements for Cheng and Church algorithm

Cheng and Church algorithm is a greedy method essentially. Because the greedy method may not always lead to correct results, we use an additional course to avoid deleting "good" rows or columns. The steps for the node addition in the original algorithm are as follows:

1. Compute e_{iJ} (for all i), e_{Ij} (for all j), e_{IJ}, and $H(I, J)$.
2. Add the columns $j \notin J$ with

$$\frac{1}{|I|}\sum_{i\in I}(e_{ij} - e_{Ij} - e_{iJ} + e_{IJ})^2 < H(I,J).$$

3. Recompute e_{iJ}, e_{IJ}, and $H(I, J)$.
4. Add the rows $i \notin I$ with

$$\frac{1}{|J|}\sum_{j\in J}(e_{ij} - e_{Ij} - e_{iJ} + e_{IJ})^2 < H(I,J).$$

5. For the i^{th} row that is still not in I, add its inverse if

$$\frac{1}{|J|}\sum_{j\in J}(e_{ij} - e_{Ij} - e_{iJ} + e_{IJ})^2 < H(I,J).$$

6. If no node needs to add in the current iteration, return the final I and J.

Considering that the search space of Cheng and Church algorithm is only a subspace of the result set, we make some improvements as follows. In order to maximize the size of the result submatrix, we amend the decision condition in the original algorithm by changing the original constraint

$$\frac{1}{|I|}\sum_{i\in I}(e_{ij} - e_{Ij} - e_{iJ} + e_{IJ})^2 < H(I,J) \quad (5)$$

into the following form

$$\frac{1}{|I|}\sum_{i\in I}(e_{ij} - e_{Ij} - e_{iJ} + e_{IJ})^2 < KH(I,J) \quad (6)$$

and when it is added into the former matrix, the value of H in the new matrix is less than the original value.

The improved algorithm extends the searching scope and increases the number of the nodes that can be added into the cluster. Our improved algorithm introduces a parameter K in Equation (6) to ensure that the blind search is reduced. Cross validation shows that the improved algorithm performs better when K is taken as 3.2.

To speed up the improved algorithm, we first express the matrix by using the idea of chromosome used in evolutionary computation, and then change the chromosomes in the matrix into two-dimensional link lists, in which we calculate the value of H and save the values of the chromosomes in the field H.

To examine the efficiency of the improved algorithm, we tested it using the yeast gene expression spectrum from the gene expression data set (*2*). For the ten clusters obtained in the simulation, we calculated the average computational time of the results and found that the original algorithm cost 65 s, while the improved algorithm cost 94 s. The comparison results are shown in Table 1. The quality of result sets of the improved algorithm is enhanced obviously, on the tolerable condition that the time cost is increased by less than 0.5 times. The comparison of the submatrix rows and columns from the improved algorithm and the result sets of the original algorithm are also shown in Table 1. It is obvious that from the results in the same condition, the improved algorithm can obtain better result sets and more information.

Table 1 Comparisons of Original and Improved Algorithms*

	A	B	C	E_B/E_A
ΔX	15.2	18.9	12.4	1.24
ΔY	2.3	2.7	1.8	1.17
ΔX+ΔY	17.5	21.6	/	1.23

*A represents the result set from the original algorithm, B is the result set from the improved algorithm, C is the intersection of A and B, ΔX is the row increment, ΔY is the column increment, and E_B/E_A is the efficiency ratio of the improved and original algorithms.

Parameter selection

In Cheng and Church algorithm, there are two important parameters δ and α that need to be set before the algorithm running, where δ is a threshold of score function H and measures the extent of data consistency. The parameter δ influences the quality of matrix clustering and in general it is better if the value is smaller. But if δ is too small, the scale of the submatrix will be over small and easy to lose information. Hence, a balance point should be found for this parameter before running the algorithm. The parameter α is used in the deletion course of the first phase in the original algorithm, which is also an important threshold. It directly influences the clustering speed. We determined the value ranges through experiments to provide referable information for realizing adaptive setting for the parameters.

Firstly, we chose real data sets for testing. Through a series of numerical experiments, we obtained the relation between the values of parameter δ and submatrix size, as shown in Figure 1. The arithmetic average of space size is used to estimate the quality of the clustering.

In the experiments, it was found that the size of the submatrix decreases monotonously with the descent of the value of δ. When δ is taken as around 120, the trend of the descent is gentle. Even the value of δ goes down again, this trend does not change essentially. Therefore, we suggest that for these data sets, it is better to take the value of δ in the range of [120, 180].

For the same data sets, we took the difference of the two systems' clocks before and after the experiment as the time consumption of clustering, and only calculated the consuming time during the course of deletion. In this way we obtained the relation between the value of α and the time consumption (Figure 2). The value of α was taken in the range of [2.8, 3.2]. We also need to make practical clustering test at $\alpha =$ 2.8 to avoid any misvalue. If the clustering results are satisfied, we could keep the range, otherwise we have to increase the lower limit.

For different data sets, we can obtain a series of results. Through linear regression and suitable adjustment to these calculation results, we have the following selection suggestions for the two parameters.

Let $[a, b]$ be the value range of data sets, m the number of genes, and n the number of conditions.

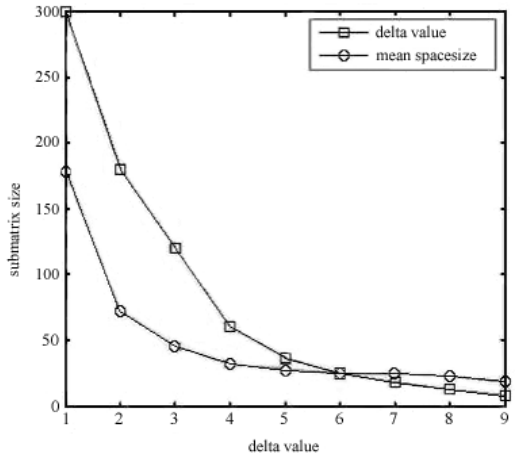

Fig. 1 The relation between δ and submatrix size.

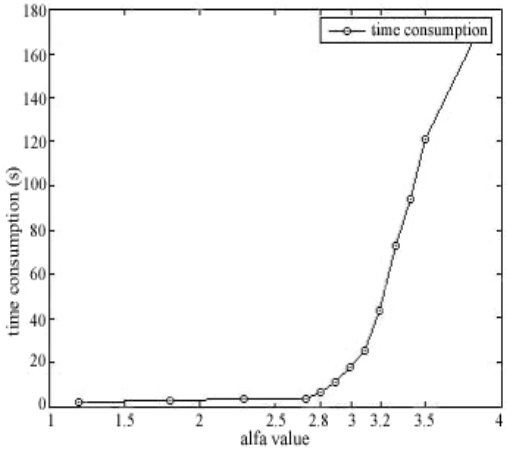

Fig. 2 The relation between α and time consumption.

Let $J = \frac{|b-a|}{mn} c_1$, where $c_1 = 5000$. The simulating experiments showed that it is better to take the value of δ in the range of $[3J, 4J]$. Let $L = \frac{|b-a|}{mn} c_2$, where $c_2 = 30$. The results of simulations showed that it is better to take the value of α in the range of $[7L, 8L]$.

Application

The open human lymphoma B cell data set (*8*) was used to examine the proposed improved algorithm. Figure 3 shows the results obtained by using the improved algorithms. Compared with the original algorithm, the quality of clustering results using the improved algorithm is enhanced obviously, the mining expression models are better, and the data possess stronger consistency with fluctuation on the condition that the time cost is increased a bit. In addition, in spite that the noise level of data sets is very high of having the loss rate of 12.3%, simulation results showed that the improved algorithm could still keep a good clustering effect even if the noise interference is very strong.

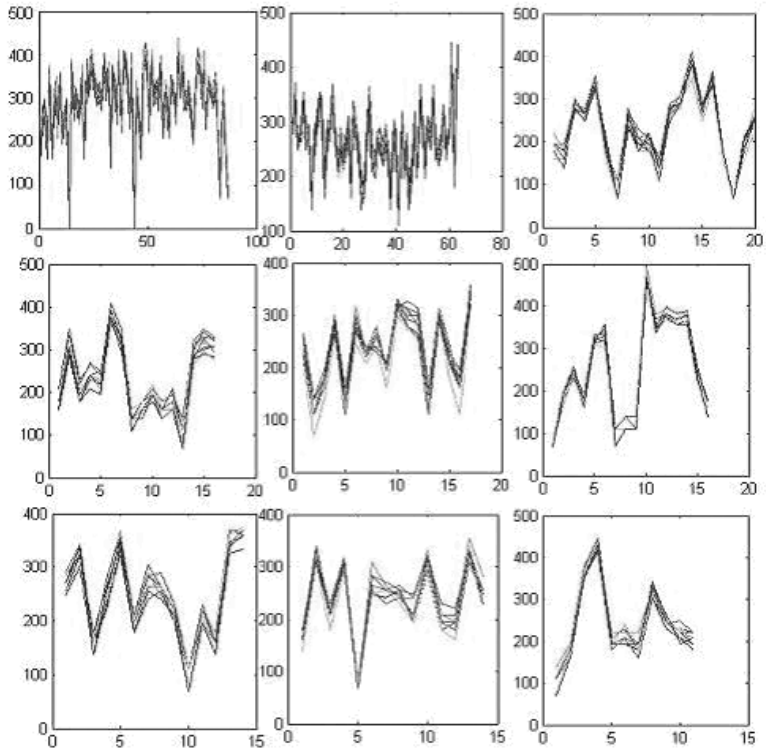

Fig. 3 Clustered result sets for the open human lymphoma B cell data using the improved algorithm.

Conclusion

The process of the extended space in the second stage of Cheng and Church algorithm is improved. The numbers of rows and columns are increased about 20% by using the improved algorithm. The effects of the two important parameters on the speed of the algorithm and the clustering quality are discussed. On the basis of simulated experiments, the experienced values for selecting parameter ranges are proposed. Real data sets with noise are used to examine the algorithms. Experimental results show the efficiency and antinoise ability of the improved algorithm.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 60433020), the Doctoral Funds of the Ministry of Education of China (No. 20030183060), the Science-Technology Development Project of Jilin Province of China (No. 20050705-2), and the "985" Project of Jilin University.

References

1. Wang, H., *et al.* 2002. Clustering by pattern similarity in large data sets. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, pp. 394-405. ACM, New York, USA.
2. Tanay, A., *et al.* 2004. Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proc. Natl. Acad. Sci. USA* 101: 2981-2986.
3. Golub, T.R., *et al.* 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* 286: 531-537.
4. DeRisi, J., *et al.* 1996. Use of a cDNA microarray to analyse gene expression patterns in human cancer. *Nat. Genet.* 14: 457-460.
5. Ben-Dor, A., *et al.* 1999. Clustering gene expression patterns. *J. Comput. Biol.* 6: 281-297.
6. Cheng, Y. and Church, G.M. 2000. Biclustering of expression data. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 8: 93-103.
7. Sheng, Q., *et al.* 2003. Biclustering microarray data by Gibbs sampling. *Bioinformatics* 19: ii196-ii205.
8. Alizadeh, A.A., *et al.* 2000. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature* 403: 503-511.