Analysis of Biclustering Algorithm using Synthetic Data

S Guru
Department Computer Science
Ayya NadarJanaki Ammal College
Sivakasi, India
guruttl90@gmail.com

T Marimuthu
Department of Computer Applications
Ayya NadarJanaki Ammal College
Sivakasi, India
mastersvksmca@gmail.com

R Lawrance
Department of Computer Applications
Ayya NadarJanaki Ammal College
Sivakasi, India
lawrancer@yahoo.com

Abstract-Biclustering technique is an alternate approach for standard clustering methods, which helps to identify the local structures from the gene expression data. These local structures provide information about main biological functions that linked with the physiological states. Biclustering method clusters the rows and column concurrently. Most of the biclustering algorithm works based on the various scores like mean square residue, variance, co-variance etc. For calculating these scores many of the algorithms follow Cheng and Church algorithm. In this work, we review the Cheng and Church algorithm and demonstrate the working procedure of the same. We have formed the synthetic data for showing the results of Cheng and Church algorithm. The results of this work clearly derived the constant and additive bicluster patterns.

Keywords- Biclusters, Mean Square Residue (MSR), threshold.

I. INTRODUCTION

A gene is the basic physical and functional unit of heredity. Genes are made up of Deoxyribonucleic acid (DNA), act as instructions to make molecules called proteins[1]. The genetic code stored in DNA is "interpreted" by gene expression. Gene expression data is obtained by extraction of quantitative information from the images/patterns resulting from the readout of fluorescent or radioactive hybridizations in a microarray chip [1]. DNA microarray is also called as DNA chip or bio chip which could be used to measure the expression levels of gene simultaneously. The DNA chip uses various techniques to generate the gene expression data. The information provided by the gene in the gene expression data is used to understand the working process. There are several steps involved to process the gene expression. They are: modulation, transcription control, and RNA splicing, translation and post translational modification of a protein.

In gene expression analysis, there are several methods used to measure the expression levels of thousands of genes over many experimental conditions (e.g., different patients, tissue types and growth environments) [2].The gene expression data could be in the form of data matrix (K × L) which contains K number of genes in row and L number of conditions in column.

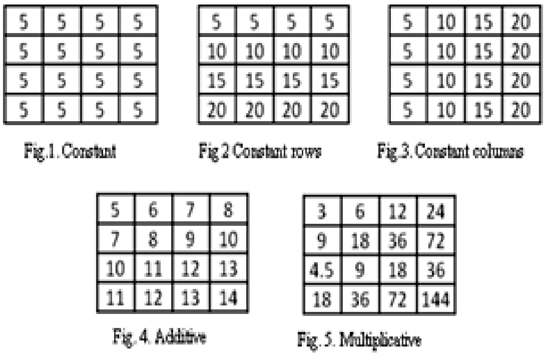
Biclustering algorithms are techniques which cluster k × l data matrix O simultaneously where K represents the rows and L represents the columns illustrated in table I. After applying biclustering technique on the data, the results obtained are known as biclusters. The biclustersO'_{kl} are the sub matrices obtained from the data where k and l are sets of row and column indices of O shown in table I which represent the subset of genes and subset of conditions. The discovery of bicluster from the gene expression data, find the regulatory patterns or condition similarities. Biclustering on gene expression data was introduced by [3], using this algorithm is not to find the maximum bicluster but main aim to find the interesting pattern corresponding genes (row) striking group of conditions (column). The rest of section in this paper could be arranged as follow. In section 2, discusses type of biclusters. In section 3, exhibit the biclustering method. In section 4, result will be discussed. Finally, we draw conclusion in section 5.

II. BICLUSTERING TYPES

Generally bicluster is a subset of genes and conditions. Bicluster can be classified in the following types based on the on the patterns exhibited by the underlying sub matrices [4].

$$\text{Bicluster}\begin{cases}\text{Having constant values, Figure 1}\\ \text{Having constant rows and columns, Figure 2 \& 3}\\ \text{Having additive based model, Figure 4}\\ \text{Having multiplicative based model, Figure 5}\end{cases}$$

[1] Http://ghr.nlm.nih.gov/handbook/basics/gene



5	5	5	5
5	5	5	5
5	5	5	5
5	5	5	5

Fig.1. Constant

5	5	5	5
10	10	10	10
15	15	15	15
20	20	20	20

Fig.2 Constant rows

5	10	15	20
5	10	15	20
5	10	15	20
5	10	15	20

Fig.3. Constant columns

5	6	7	8
7	8	9	10
10	11	12	13
11	12	13	14

Fig. 4. Additive

3	6	12	24
9	18	36	72
4.5	9	18	36
18	36	72	144

Fig. 5. Multiplicative

Figure 1. Types of Bicluster

III. PROCEDURE

Now we can analysis the data using the Cheng and Church (CC) algorithm [3]. To analyze the effectiveness of the method, we can use synthetic dataset (See Table 1).

A. *Dataset*

Real gene expression data contain lots of noise and the biclustering algorithms may not be able to extract all the bicluster contain in the data. Information hidden in the results could be hard to define. Therefore, we use synthetic data for our empirical study.

We represent the k × l matrix O shown in table I. The real number entry o_{kl} represent gene data set whose K represents the number of genes and L represents the number of conditions. It can be a relative expression ratio (eg., cDNA microarrays) otherwise it can be an absolute value (eg.,AffymetrixGeneChip) [10].

TABLE I. SYNTHETIC DATASET

	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10
r1	12	2	24	98	48	75	59	47	77	26
r2	27	44	5	78	7	31	4	23	81	2
r3	6	55	12	55	24	55	48	55	71	31
r4	28	17	6	97	8	58	5	86	92	3
r5	18	55	36	55	72	55	25	55	74	97
r6	76	18	8	21	10	78	7	97	70	5
r7	30	55	64	55	50	55	92	55	23	36
r8	17	30	10	59	12	89	9	61	52	7
r9	77	66	31	50	95	56	92	78	20	44
r10	14	12	7	4	9	63	6	92	64	4

It shown table I is used in the describing the process of the CC algorithm [2].

B. *Biclustering Algorithm Workflow*

We use cc algorithm [3] was proposed by Cheng and Church in 2000 the algorithm is based on a simple uniformity goal which is the Mean Square Residue (MSR) [6]. A greedy approach to find the largest bicluster from that bicluster it combined iteratively to find more biclusters. To discover the largest δ bicluster from the whole data is NP hard. A naïve greedy algorithm used for detecting the δ bicluster from the given data and it also apply the brute force technique to each and every single row and column by adding or deleting them, this could be terminates until finding the improved score, that is bicluster score is below a certain δ threshold value [7].

Given a k × l data matrix O with set of K rows and L columns, we indicate the matrix as Ok,l We could calculate the following value to define a bicluster (K, L),

Row subset average,

$$o_{kL} \;=\; \frac{1}{|l|}\sum_{l\in L} o_{kl} \tag{1}$$

Column subset average,

$$o_{Kl} \; = \frac{1}{|k|} \sum_{k \in K} o_{kl} \tag{2}$$

Sub Matrix average,

$$o_{KL} = \frac{1}{|K|\,|L|} \sum_{k \in K, l \in L} o_{kl} = \frac{1}{|K|} \sum_{k \in K} o_{kL} = \frac{1}{|L|} \sum_{l \in L} o_{Kl} \tag{3}$$

With this residue score

$$r(o_{k,l}) = o_{kl} - o_{Kl} - o_{kL} + o_{KL} \tag{4}$$

MSR score,

$$H(K, L) = \frac{1}{(|K||L)} \sum_{k \epsilon K, l \epsilon L} \left(r(o_{k,l})\right)^2 \tag{5}$$

To calculate the row score,

$$d(k) = \frac{1}{|L|} \sum_{l \epsilon L} r(o_{k,l}) \tag{6}$$

Column score,

$$e(l) = \frac{1}{|K|} \sum_{k \epsilon K} r(o_{k,l}) \tag{7}$$

The above equations are applied to the data then the values are used to discover the bicluster pattern, the following pseudo code describes the working procedure of the CC algorithm.

1) Single Node Deletion Algorithm

Input:

- Data matrixO'_{kl}**,** k refer row, l refer the column and a parameter δ

Process:

- Using equation (1),(2),(3) to calculate the row (column) subset averageand overall average
- Then using equation (5) to calculate the MSR score.
- If MSR score value is greater than **δ** threshold value, produce output (K, L).
- The row score using equation (6) is calculated for each row
- The column score using equation (7) is calculated for each column
- Select the row and column which has score value greater than δ threshold, otherwise eliminate the row and column from the bicluster

Result:

- $O_{K,L}$ a δ-bicluster

2) Multiple Node Deletion Algorithm

Input:

- Data matrix O'_{kl}, δ ≥ 0, α threshold for multiple node deletion

Process

- Calculate MSR score for sub matrix O'_{kl},
- Using equation (2) we calculate the column subset average,
- Eliminate row If row subset score greater than α times the MSR score
- Repeat the iteration,
- Same process for column elimination, here we use equation (1) to calculate row subset average
- If no elimination done, the use single node deletion algorithm.

3) Node addition algorithm

Input:

- Data matrixO'_{kl}, parameter δ, K × L specifying a δ-bicluster

Process:

- Calculate the row (column) subset average and MSR score.
- Check the column score, if column score is less than the MSR score, then corresponding column is added

- Using equation (1), (2), (5) recalculate the row mean, column mean, and MSR score.
- Check the row score, if row score less than MSR score, then corresponding column is added
- Algorithm quit, when adding process is stopped.

Result:

- $O_{K', L'}$ a δ-bicluster, with K' subset of K and L' subset of L

4) *Cheng and Church algorithm*

Input:

- Data matrix **O,** parameter **δ,** m represents the number bicluster needed

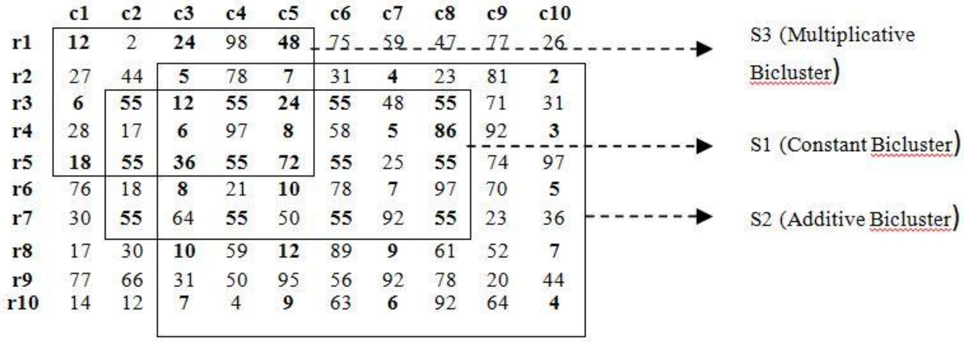
Process:

- Does deletion process multiple times on O produce K' × L' bicluster
- Using node addition algorithm to the K' × L' produce O'_{kl}

Result:

- m number of bicluster obtain is less than δ threshold value

By applying the CC algorithm [3] to, a synthetic data (shown in table I) having three overlapping biclusters. They are constant bicluster S1={r3,r5,r7}×{c2,c4,c6,c8}, additive bicluster S2={r2,r4,r6,r8,r10}×{c3,c5,c7,c10}, multiplicative bicluster S3 = {r1,r3,r5}×{ c1,c3,c5}. The overlapped biclusters present in synthetic data.



	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10
r1	**12**	2	**24**	98	**48**	75	59	47	77	26
r2	27	44	**5**	78	**7**	31	**4**	23	81	**2**
r3	**6**	**55**	**12**	**55**	**24**	**55**	48	**55**	71	31
r4	28	17	**6**	97	**8**	58	**5**	**86**	92	**3**
r5	**18**	**55**	**36**	**55**	**72**	**55**	25	**55**	74	97
r6	76	18	**8**	21	**10**	78	**7**	97	70	**5**
r7	30	**55**	64	**55**	50	**55**	92	**55**	23	36
r8	17	30	**10**	59	**12**	89	**9**	61	52	**7**
r9	77	66	31	50	95	56	92	78	20	44
r10	14	12	**7**	4	**9**	63	**6**	92	64	**4**

S3 (Multiplicative Bicluster)
S1 (Constant Bicluster)
S2 (Additive Bicluster)

Figure 2. Bicluster

we calculate the mean of the each row using equation (1), second to calculate the mean of the each column equation (2), finally calculate the overall mean of the data matrix equation (3), then we get the MSR score equation (5), the calculated MSR value for synthetic data is 515.1001

Using equation (6) to calculate the row score, the values for each row k1 is 447.5709, k2 is 444.6629, k3 is 155.0429, k4 is 543.8469, k5 is 640.7229, k6 is 680.6069, k7 is 686.3469, k8 is 202.6949, k9 is 931.9789, and k10 is 417.5269.

Using equation (7) to calculate the column score, the values for each column are l1 is 503.5549, l2 is 314.1269, l3 is 156.2989, l4 is 781.4509, l5 is 420.9149, l6 is 351.1949, l7 is 708.5709, l8 is 590.7469, l9 is 863.3469, and l10 is 460.7949.

Row (column) addition, deletion belongs to the row score and the column score, this will be clearly described in section III. The row score and the column score could be changed during the addition and the deletion phase. Note δ threshold value is greater than or equal to zero.

$$H(K,L) \begin{cases} \geq \delta \text{ threshold value result is whole data} O_{kl} \\ _{<}\delta \text{ threshold value } O'_{kl} \text{ (constant, additive model is resulted)} \end{cases} \tag{8}$$

IV. Result and Discussion

By passing user defined δ threshold value 1 and set α value 0 , then we get the following bicluster pattern , B1 = {r2, r4, r6, r8, r10} × {c3, c5, c7, c10}, B2 = {r3, r5, r7} × {c6, c8}, B3 = {r1, r9} × {c7, c8}.

From the result, we got three biclusters shown in table II, III & IV, from our synthetic data, we could compare the result with three overlapped bicluster (S1, S2, S3) are already embedded in synthetic data . From our result, bicluster already embedded in synthetic data S2 (additive bicluster) equal to B1 (additive bicluster extract using CC[3] method) and constant bicluster pattern embedded in synthetic data (S1) which is not fully extracted only partially extracted during the process shown in Table III. Multiplicative pattern S3 is not extracted, that implies multiplicative pattern is not fits for CC [3] method.

TABLE I. ANALYSIS OF CHEN AND CHURCH ALGORITHM USING DIFFERENT THRESHOLD VALUE

For threshold value (δ, α)	Total bicluster	Constant Bicluster	Additive Type	Average Mean Square Residue (AMSR)
1,0	3	1	2	0.083333
10,0	3	1	2	0.083333
50,0	2	0	2	38.10815
100,0	2	0	2	64.73958
150,0	2	0	2	112.4125

From the above Table V, clearly shows that AMSR value is less than user defined δ value.

Applying CC algorithm [3] on the synthetic data set shown in table I. After analysis with various threshold value δ shown in table V CC algorithm [3] is well suitable for constant and additive based biclusters and it is not fits for finding the multiplicative model.

Applying the MSR score to search for biclusters, some high-quality biclusters will be missed [8]. One notable drawback, however, of the MSR score is that it is also affected by variance [2]. Correlated patterns are found only when variance value is low, the variance value is high in MSR score is high, therefore coherent bicluster patterns is not found by applying cc algorithm [3].

CC algorithm tends to generate large biclusters that often represent gene groups with unchanged expression levels. Therefore interesting patterns in terms of co-regulation are not necessarily contained by [6]

The quality of bicluster is bad if user defined threshold value should increased. It clear from the analysis shown in table V increase in the AMSR value will decrease in pattern discovery.

V. CONCLUSION

The concept of biclustering was mainly introduced for analyzing the gene expression and it was introduced by Cheng and Church. The CC biclustering algorithm worked on basic MSR score. Discovering the interesting patterns on using MSR score is difficult to identify the co-regulated patterns. This is because the variance value associate with MSR is very high. Co regulated patterns (see figure 2.5) favoring in low variance. In this paper, it has been clearly shows working procedure of the CC algorithm[2], by using a synthetic data set shown in table I it defines that biclusters are discovered based on the MSR score, also proves the importance in improving the score to identify quality bicluster patterns. The result of this work will help the newcomer’s in field of biclustering and also they know about the importance in discovering the patterns from gene expression data.

REFERENCES

[1] Emilyn, J. J., &Ramar, K. Rough set based clustering of gene expression data: A survey. Int. J. of Eng. Sci. and Technol, vol-2, no-12, pp. 7160-7164, 2010.

[2] Hussain, S., & Hazarika, G. Improved Biclustering Of Microarray Data. Journal of Computer and Mathematical Sciences, vol-1, no-2, pp. 103-273, 2010.

[3] Cheng, Y., & Church, G. M. Biclustering of expression data, Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology, vol, 8, pp. 93-103, 2000.

[4] Parimala, R. "Enhanced Biclustering for Gene Expression Data.", International Journal of Science and Modern Engineering, vol-1, no-5, 2013.

[5] Yang, W. H., Dai, D. Q., & Yan, H. Finding correlated biclusters from gene expression data. Knowledge and Data Engineering, IEEE Transactions on, vol-23, no-4, pp. 568-584, 2011.

[6] Prelić, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W.,&Zitzler, E.”A systematic comparison and evaluation of biclustering methods for gene expression data”,. Bioinformatics, vol-22, no-9, 2006, pp. 1122-1129.

[7] Gremalschi, S., &Altun, G. Mean squared residue based biclustering algorithms. In Bioinformatics Research and Applications. Springer Berlin Heidelberg, pp. 232-243, 2008.

[8] Aguilar-Ruiz, J. S. (2005). Shifting and scaling patterns from gene expression data. Bioinformatics, vol, 21, no-20, pp. 3840-3845.