Biclusters Visualization and Detection Using Parallel Coordinate Plots

K.O. Cheng[1], N.F. Law[1], W.C. Siu[1] and A.W.C. Liew[2]

[1]Centre for Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong.
[2]School of Information and Communication Technology, Griffith University, Gold Coast Campus, QLD 4222, Queensland, Australia.

Abstract. The parallel coordinate (PC) plot is a powerful visualization tools for high-dimensional data. In this paper, we explore its usage on gene expression data analysis. We found that both the additive-related and the multiplicative-related coherent genes exhibit special patterns in the PC plots. One-dimensional clustering can then be applied to detect these patterns. Besides, a split-and-merge mechanism is employed to find the biggest coherent subsets inside the gene expression matrix. Experimental results showed that our proposed algorithm is effective in detecting various types of biclusters. In addition, the biclustering results can be visualized under a 2D setting, in which objective and subjective cluster quality evaluation can be performed.

Keywords: Biclustering, bioinformatics, clustering, gene expression data analysis.
PACS: 87.15.Aa, 89.75.Kd, 89.20.Ff

INTRODUCTION

Data from microarray experiments are usually expressed as a large matrix containing gene expression levels (rows) under different experimental conditions (columns). One of the challenges in gene expression data analysis is to identify co-expressed genes which exhibit similar behavior. Traditional clustering techniques are global in nature in which grouping is performed along the entire rows or across the entire columns [1, 2]. In practice, genes co-express only under certain experimental conditions. Biclustering which perform clustering simultaneously along row and column directions is thus highly desired [1, 3, 11, 12].

Besides biclustering, visualization of the gene expression data is often helpful for analysis. However, the visualization is not trivial due to the high dimensional nature [4]. One of the powerful tools for visualizing high-dimensional data is the parallel coordinate (PC) plot in which each dimension is represented as a vertical axis, and the *N*-dimensional axes are drawn in parallel to each other [5, 6]. In this paper, the PC technique is proposed to detect and visualize biclusters embedded in the gene expression matrix. First, special patterns of different types of biclusters are investigated using the PC plots. Then, the biclustering problem is reformulated as finding these special patterns in the PC plots. Experimental results will be presented to show the effectiveness of the proposed biclustering algorithm.

BICLUSTER ANALYSIS USING THE PC PLOT

In a gene expression matrix, rows represent genes while columns represent experimental conditions. A bicluster is a subset of rows which exhibit coherent patterns across a subset of columns. There are two general types of biclusters, namely additive-related and multiplicative-related biclusters [1]. Figure 1 shows examples of these two types of biclusters.

C1	C2	C3
1	5	2
3	7	4
5	9	6

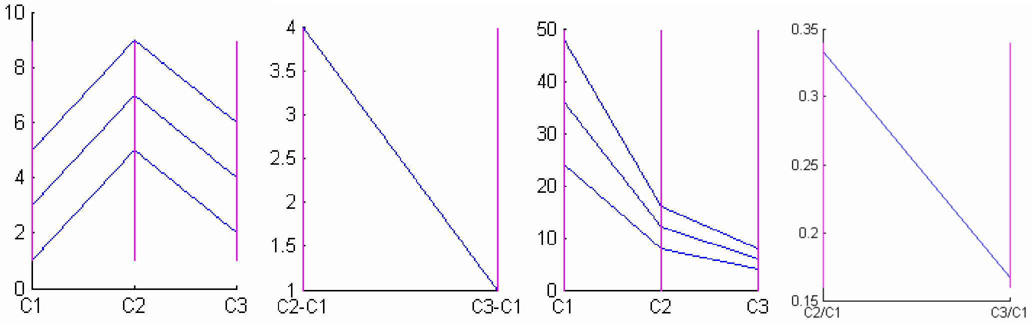
(a)

C1	C2	C3
48	16	8
24	8	4
36	12	6

(b)

FIGURE 1. (a) An additive-related bicluster and (b) a multiplicative-related bicluster.

A way to visualize the high dimensional data is to use the parallel coordinate (PC) plot. All axes are arranged in parallel to each other on a 1D plane. Despite the fact that the orthogonal property is destroyed, geometric structure can still be preserved by the PC plot [5, 6]. Figure 2(a) shows the PC plot of the additive-related bicluster of Figure 1(a). We can see that the additive-related bicluster shows a number of lines with the same slope across the conditions. Thus if {C2-C1, C3-C1} is considered as in Figure 2(b), the additive-related bicluster can be identified as a single clustered point. For the multiplicative-related bicluster in Figure 1(b), direct PC plot in Figure 2(c) does not show any simple structure. However, if {C2/C1, C3/C1} is considered in the PC plot as in Figure 2(d), an overlapped line is obtained in which the multiplicative-related bicluster can be identified again as a single clustered point. Based on these observations, the problem of biclusters identification can be reformulated as finding these special structures, i.e., clustered points, in the PC plots.



(a) (b) (c) (d)

FIGURE 2. PC plots for (a) an additive-related bicluster; (b) the bicluster in (a) with axes {C2-C1, C3-C1}; (c) a multiplicative-related bicluster and (d) the bicluster in (c) with axes {C2/C1, C3/C1}.

THE PROPOSED BICLUSTERING ALGORITHM

Although biclusters appear as clustered points in the PC plots, their detection is complicated as rows and columns in a bicluster may not be in a consecutive order in a gene expression matrix. The presence of unrelated rows and unrelated columns obscures those special structures and thus biclusters stay hidden. To solve this problem, every two columns are compared so as to identify the related columns first. Here, the related columns mean the existence of a clustered point as in Figure 2(b) and Figure 2(d). To illustrate the idea, let us consider the data shown in Figure 3. In searching for the clustered points, a difference matrix, i.e., the differences between two columns, is formed as in Figure 4. Consider column “*C*5-*C*3”. There are only three distinct clustered points: 0 (5 counts), 1 (1 count), 2 (5 counts). This suggests the existence of three biclusters between “*C*5” and “*C*3”.

- the first bicluster is for rows *R*1, *R*3, *R*5, *R*9 and *R*11 in which the difference between “*C*5” and “*C*3” is zero, i.e., a constant bicluster;
- the second bicluster is for rows *R*2, *R*4, *R*6, *R*8 and *R*10 in which the difference between “*C*5” and “*C*3” is two, i.e., an additive bicluster;
- the third bicluster involves row *R*7 only, thus it is not a valid bicluster.

Thus by merging “*C*3” and “*C*5”, two biclusters are formed as in Figure 5. The analysis can be repeated for each of these two groups to find out whether any other columns can be merged to {*C*3, *C*5}, i.e., using either *C*3 or *C*5 as a reference, check whether *C*1, *C*2, *C*4, and *C*6 can be merged with {*C*3, *C*5}. As in Figure 6, two difference matrices are obtained. Note that the difference values can be read directly from the original difference matrix of Figure 4. By examining the first difference matrix in Figure 6, we can see that two paired columns, “*C*1-*C*3” and “*C*2-*C*3”, show a single clustered point with the difference value equals to zero. This suggest that columns “*C*1’ and “*C*2” can be merged to {*C*3, *C*5} for rows *R*1, *R*3, *R*5, *R*9 and *R*11. The second difference matrix also has a single clustered point with value equal to 1. Therefore, “*C*6” can be merged to {*C*3, *C*5} for rows *R*2, *R*4, *R*6, *R*8 and *R*10. By this repeated “merge and split” process – merging the paired columns and splitting the rows, we can identify possible biclusters embedded in the dataset.

	C1	*C2*	*C3*	*C4*	*C5*	*C6*
R1	1	1	1	5	1	0
R2	1	3	2	2	4	3
R3	1	1	1	2	1	2
R4	3	1	3	6	5	4
R5	1	1	1	0	1	3
R6	2	3	3	1	5	4
R7	0	3	6	7	7	1
R8	4	5	2	1	4	3
R9	1	1	1	3	1	3
R10	6	0	1	6	3	2
R11	1	1	1	2	1	4

FIGURE 3. A dataset consists of two biclusters.

	C2 - C1	C3 - C1	C4 - C1	C5 - C1	C6 - C1	C3 - C2	C4 - C2	C5 - C2	C6 - C2	C4 - C3	C5 - C3	C6 - C3	C5 - C4	C6 - C4	C6 - C5

R1	0	0	4	0	-1	0	4	0	-1	4	0	-1	-4	-5	-1
R2	2	1	1	3	2	-1	-1	1	0	0	2	1	2	1	-1
R3	0	0	1	0	1	0	1	0	1	1	0	1	-1	0	1
R4	-2	0	3	2	1	2	5	4	3	3	2	1	-1	-2	-1
R5	0	0	-1	0	2	0	-1	0	2	-1	0	2	1	3	2
R6	1	1	-1	3	2	0	-2	2	1	-2	2	1	4	3	-1
R7	3	6	7	7	1	3	4	4	-2	1	1	-5	0	-6	-6
R8	1	-2	-3	0	-1	-3	-4	-1	-2	-1	2	1	3	2	-1
R9	0	0	2	0	2	0	2	0	2	2	0	2	-2	0	2
R10	-6	-5	0	-3	-4	1	6	3	2	5	2	1	-3	-4	-1
R11	0	0	1	0	3	0	1	0	3	1	0	3	-1	2	3

FIGURE 4. The difference matrix for the dataset in Figure 3.

	{*C*3, *C*5}	*C*1	*C*2	*C*4	*C*6
*R*1	{1, 1}	1	1	5	0
*R*3	{1, 1}	1	1	2	2
*R*5	{1, 1}	1	1	0	3
*R*9	{1, 1}	1	1	3	3
*R*11	{1, 1}	1	1	2	4

	{*C*3, *C*5}	*C*1	*C*2	*C*4	*C*6
*R*2	{2, 4}	1	3	2	3
*R*4	{3, 5}	3	1	6	4
*R*6	{3, 5}	2	3	1	4
*R*8	{2, 4}	4	5	1	3
*R*10	{1, 3}	6	0	6	2

FIGURE 5. The two different groups formed by merging columns "*C5*" and "*C3*".

	{*C*3, *C*5}	*C*1-*C*3	*C*2-*C*3	*C*4-*C*3	*C*6-*C*3
*R*1	{1, 1}	0	0	4	-1
*R*3	{1, 1}	0	0	1	1
*R*5	{1, 1}	0	0	-1	2
*R*9	{1, 1}	0	0	2	2
*R*11	{1, 1}	0	0	1	3

	{*C*3, *C*5}	*C*1-*C*3	*C*2-*C*3	*C*4-*C*3	*C*6-*C*3
*R*2	{2, 4}	-1	1	0	1
*R*4	{3, 5}	0	-2	3	1
*R*6	{3, 5}	-1	0	-2	1
*R*8	{2, 4}	2	3	-1	1
*R*10	{1, 3}	5	-1	5	1

FIGURE 6. The two difference matrix formed by merging columns "*C5*" and "*C3*".

RESULTS AND DISCUSSION

We analyze the performance of our algorithm on both synthetic and real datasets. As gene expression values are often corrupted by noise, we will first investigate the performance of our algorithm on noisy synthetic data. Then, we will present our result on a real dataset: the yeast "Saccharomyces cerevisiae" cell cycle data.

Noisy Artificial Dataset

The dataset is of dimension 100 by 10. Its values are uniformly distributed between -5 and 5. A bicluster pattern of 30 rows by 4 columns is embedded in which related columns are randomly placed. The bicluster is an additive-related pattern with Gaussian noise of variance equal to 0.2. There are two thresholds to be set in our algorithm. The first is the noise threshold which is used to define the similarity of expression values and used in the 1D clustering among paired columns. The second threshold is the minimum number of rows that should be maintained when columns are merged. This threshold is important as it prevents column merging if the number of rows falls below the threshold after merging.

Figure 7 shows a bicluster found by our algorithm when the noise threshold is set to be 1.2. The four columns are identified correctly, but, three rows are missed. Figure 8 shows the four columns: the rows that are found are displayed in red, while the three missed rows are displayed in blue. We can see that these three missing rows are caused by the use of a small noise threshold. In practice, we do not know the bicluster in advance. We can adopt an exploratory approach for setting the appropriate noise threshold [6]. Starting with a small noise threshold, we gradually increase its value while visualizing the detected bicluster using the PC plot. By increasing the noise threshold, more rows are included in the bicluster. Then, at some point, unrelated rows start to creep into the bicluster. Once this occurred and is observed in the PC plot, we stop increasing the noise threshold. Using this procedure, we found that when the threshold is set to 1.5, all the rows are correctly detected. This example shows that the PC plot can be a powerful visualization and interactive tool that allows us to examine the quality of the detected bicluster.

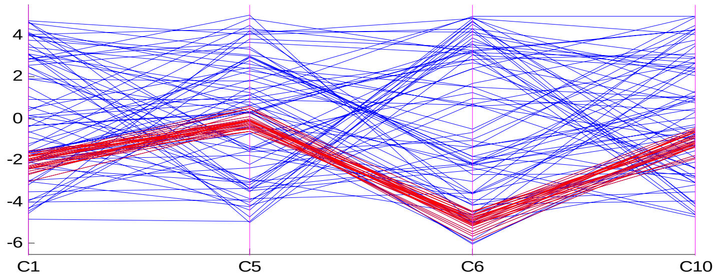


FIGURE 7. The PC plot of the four related columns in the additive related bicluster. Red color shows rows from the true bicluster while blue color shows rows from the original dataset

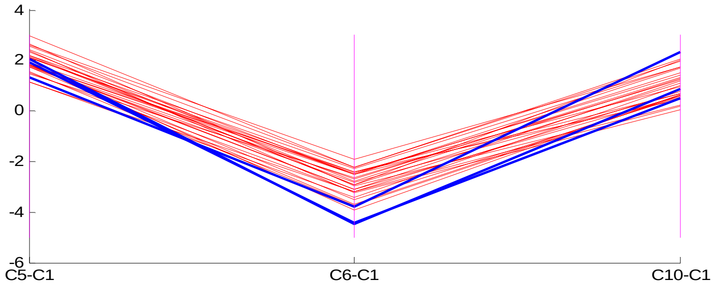


FIGURE 8. The PC plot of the difference between the last three columns and the first column. The red color shows rows of the true bicluster that are found by our algorithm with noise threshold = 1.2 while the blue color shows the three rows of the true bicluster that are missed out.

Real Dataset – Yeast S. Cerevisiae

This dataset describes the cell cycle expression of S. cerevisiae [8]. It contains 2884 genes and 17 conditions. By setting the minimum number of genes and the noise threshold to be 20 and 5 respectively, 100 biclusters are found. Table I shows the mean square residue score (MSRS) [8] and average correlation value (ACV) [9] of the biclustering results. A highly homogeneous bicluster should have a low MSRS and a high ACV. Over the 100 detected biclusters, the average MSRS has a small value equal to 0.01553 and the average ACV has a high value equal to 1. Besides, we have found that the biggest bicluster also exhibits the lowest MSRS and the highest ACV at the same time. Thus, the proposed algorithm can detect significant homogeneous biclusters.

Table I. A summary of biclustering results using the proposed algorithm

Bicluster	Size	MSRS	ACV
Average	-	0.01553	1
Biggest size	21x10	0	1
Least variation	21x10	0	1
Highest correlation	21x10	0	1

A main concern in the study is whether biclusters with genes having the same function can be discovered. To evaluate the ability of the proposed algorithm in finding such kind of biclusters, we calculate percentage of biclusters which are overrepesented in one or several Gene Ontology (GO) annotation. A bicluster is regarded to be overrepresented in a functional category if the probability for obtaining the category by random (*p*-value) is significantly small. We considered five types of categories including biological process, cellular component, deletion viability, molecular function and regulatory pathway with *p*-values less than 0.05. The results are generated using the software GeneMerge [10]. The results have also been compared with Cheng and Church algorithm. The number of detected biclusters in Cheng and Church algorithm is set to be the same as that found by the proposed algorithm. The results are provided in Table II.

For p-value < 0.05, the proposed algorithm shows the highest enrichment in cellular component categories. The percentage of enriched biclusters is 81%. Cheng and Church algorithm also has the highest enrichment in cellular component, but the percentage is 55% only. The proposed algorithm also outperforms Cheng and Chruch algorithm in other categories except the regulatory pathway. Besides the percentage of abundance, results for functional enrichment in cell component categories are also provided in Table III. Note that we only showed the results with the corrected p-value <0.05. For a more comprehensive set of results and the software codes, please refer to the web site: http://www.eie.polyu.edu.hk/~nflaw/Biclustering. Among the 100 biclusters, the 24-th bicluster has the lowest *p*-value, which is equal to 1.23×10^{-5}, and there are 11 genes associated with the cellular component category. Despite the fact that not all biclusters have GO terms assigned, biclusters without category associated may contain genes unknown to certain functions according to current knowledge. Further study on those biclusters may lead to new biological findings. More detail GO analysis of the detected biclusters is currently under investigation.

Table II. Comparison between the proposed biclustering algorithm and Cheng and Church algorithm in functional enrichment.

Algorithm	*p*-value	Categories*				
		BP	CC	DEL	MF	PATH
Proposed	< 0.05	65%	81%	33%	37%	11%

Cheng & Church	< 0.05	54%	55%	11%	35%	17%

* BP, CC, DEL, MF and PATH stand for biological process, cellular component, deletion viability, molecular function and regulatory pathway respectively.

Table III. Details of the functional enrichment based on GO annotation of cell component categories for both *p*-value and the corrected p-value less than 0.05.

Bicluster index	Annotation	P-value	Corrected P-value	Genes
1	condensed nuclear chromosome	1.04E-03	1.14E-02	YER179W, YPL194W
5	condensed nuclear chromosome	1.14E-05	1.49E-04	YER179W, YHR157W, YPL194W
7	condensed nuclear chromosome	1.14E-05	1.26E-04	YER179W, YHR157W, YPL194W
8	condensed nuclear chromosome	2.27E-03	2.95E-02	YHR157W, YPL194W
9	condensed nuclear chromosome, pericentric region	2.27E-03	3.63E-02	YGR188C, YHR014W
15	condensed nuclear chromosome	9.40E-04	1.32E-02	YER179W, YPL194W
	integral to membrane	3.36E-03	4.71E-02	YER060W, YFL054C, YNL194C
16	condensed nuclear chromosome	9.40E-04	1.03E-02	YER179W, YPL194W
17	condensed nuclear chromosome	1.14E-03	1.82E-02	YER179W, YPL194W
22	condensed nuclear chromosome	1.04E-03	1.45E-02	YER179W, YPL194W
38	prospore membrane	2.19E-03	3.73E-02	YER096W, YLR054C
52	condensed nuclear chromosome, pericentric region	9.40E-04	1.22E-02	YGR188C, YHR014W
57	condensed nuclear chromosome	1.25E-03	1.50E-02	YHR079C, YHR157W
81	condensed nuclear chromosome	9.40E-04	8.46E-03	YHR157W, YPL194W

CONCLUSIONS

We have investigated the use of parallel coordinate (PC) plots on biclusters detection and visualization. The special structures exhibited by biclusters in the PC plots have been studied. With the use of these special structures, 1D clustering and a split and merge mechanism have been employed for biclusters detection. Beides, the PC plots allow detected biclusters to be visualized interactively under a 2D setting. We have verified the performance of our algorithm using both artificial noisy datasets and real datasets. Experimental results showed that our proposed algorithm is effective in detecting meaningful biclusters.

ACKNOWLEDGMENTS

This work is supported by the Centre for Signal Processing, the Department of Electronic and Information Engineering and the Hong Kong Polytechnic University (A-PA2P). K.O. Cheng acknowledges the research studentships provided by the University.

REFERENCES

1. M S.C. Madeira and A.L. Oliveira, “Biclustering Algorithms for Biological Data Analysis: A Survey”, IEEE/ACM Trans. Computational Biology and Bioinformatics, vol. 1, no. 1, pp. 24-45, 2004.
2. A. Ben-Dor, B. Chor, R. Karp and Z. Yakhini, “Discovering Local Structure in Gene Expression Data: The Order-preserving Submatrix Problem”, J. Computational Biology, vol. 10, nos. 3-4, pp. 373-384, 2003.
3. A. Prelic, S. Bleuler, P. Zimmermann, A. Wille, P. Buhlmann, W. Gruissem, L. Hennig, L. Thiele and E. Zitzler, “A Systematic Comparison and Evaluation of Biclustering Methods for Gene Expression Data”, Bioinformatics, vol. 22, no. 9, pp. 1122-1129, 2006.
4. T. V. Prasad and S. I. Ahson, “Visualization of Microarray Gene Expression Data”, Bioinformation, vol. 1, pp. 141-145, 2006.
5. A. Inselberg and B. Dimsdale, “Parallel Coordinates: A Tool for Visualizing Multidimensional Geometry”, Proc. Of Visualization, pp. 361-378, 1990.
6. W. Peng, M. O. Ward and E. A. Rundensteiner, “Clutter Reduction in Multi-Dimensional Data Visualization Using Dimension Reordering”, IEEE Symposium on Information Visualization, pp. 89 - 96, Oct. 2004.
7. K.O. Cheng, N.F. Law, W.C. Siu and T.H. Lau, “BiVisu: Software Tool for Bicluster Detection and Visualization”, Bioinformatics 2007, doi: 10.1093/bioinformatics/btm338.
8. Y. Cheng and G.M. Church, “Biclustering of Expression Data”, Proceedings, Conference on Intelligent Systems for Molecular Biology, pp. 93-103, 2000.
9. L. Teng and L.-W. Chan, “Biclustering Gene Expression Profiles by Alternately Sorting with Weighted Correlated Coefficient”, Proceedings of IEEE International Workshop on Machine Learning for Signal Processing, pp. 289 – 294, 2006.
10. C.I. Castillo-Davis and D.L. Hartl, "GeneMerge - post-genomic analysis, data mining, and hypothesis testing", Bioinformatics, vol.19, no.7, pp.891-892, 2003.
11. X. Gan, A.W.C. Liew, and H. Yan, "Biclustering Gene Expression Data based on a High Dimensional Geometric Method", Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, ICMLC 2005, Guangzhou, China, 19-21 August 2005
12. H. Zhao, A.W.C. Liew, and H. Yan, "A New Strategy of Geometrical Biclustering for Microarray Data Analysis", Proceedings of the Fifth Asia Pacific Bioinformatics Conference, APBC2007, Hong Kong, 15-17 January 2007