2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

A Modified Fuzzy Co-Clustering (MFCC) Approach for Microarray Data Analysis

Sheng-Yao Huang, Hsing-Jen Sun, Chuen-Der Huang, I-Fang Chung, and Chun-Hung Su

***Abstract*—Biologically a gene or a sample could participate in multiple biological pathways, and only few genes are concurrently involved in a cellular process under some specific experimental conditions. Hence, identification of a subset of genes showing similar regulations under subsets of condition in microarray data has become an important research issue. Many investigators develop bi-clustering methods to attack this problem. In this study, we adopt fuzzy co-clustering concept and design a procedure to iteratively extract bi-clusters with co-expressed gene patterns (here the entire proposed process is called a modified fuzzy co-clustering (MFCC) approach). We have applied synthetic data and compared our MFCC's performance with four well-known state-of-the-art methods. Here we have not only shown that our MFCC approach can successfully extract each designed bi-clusters in the synthetic data sets, but also have demonstrated the better performance by our MFCC approach.**

I. Introduction

AFTER finishing the Human Genome Project, one of the most urgent and important tasks for scientists in the post-genomics era has been to understand the function of tens of thousands of genes, especially for deciphering the relationship between genes and diseases in biomedical research. Microarray (also called gene chip) is a widely used tool for measuring gene expression values in genomics research. Microarray is composed of thousands of probes which can specifically bind to their corresponding gene target based on the principle of pairing complementarity of nucleotide bases. Hence, the microarray technique can simultaneously measure expression values of a large number of genes, and some of the most important research topics in bioinformatics have been how to process and analyze microarray high-throughput data, and then further discover marker genes to investigate oncogenic factors, study the biochemical mechanisms of drug-resistance, or evaluate the elements of prognosis.

S. Y. Huang is with Institute of Biomedical Informatics, National Yang-Ming University, Taipei 11221, Taiwan (e-mail: d49623005@ym.edu.tw).

H. J. Sun is with Institute of Biomedical Informatics, National Yang-Ming University, Taipei 11221, Taiwan (e-mail: rodney7646@ gmail.com).

C. D. Huang is with Department of Electrical Engineering, Hsiuping University of Science and Technology, Taichung 41280, Taiwan (e-mail: cdhuang@mail.hust.edu.tw).

I. F. Chung is with Institute of Biomedical Informatics, National Yang-Ming University, Taipei 11221, Taiwan (e-mail: ifchung@ym.edu.tw).

C. H. Su is with the Genomics Research Center, Academic Sinica, Taipei 11529, Taiwan (corresponding author: +886-2-2787-1287; e-mail: such@gate.sinica.edu.tw).

This work is supported by National Science Council (NSC), Taiwan, Grant No.: NSC102-2221-E-010-014.

Previous statistical methods involving clustering analysis of microarray data have mostly focused on data representing the axiality of genes or axiality of experimental conditions (or samples) in order to identify the relationship among genes or the relationship among experimental conditions (or samples). However, many genes show high/low expression values only under certain specific experimental conditions, such that these genes have similar expression levels under the regulation of specific conditions but have other gene expression patterns under other conditions. Hence, many researchers have developed various bi-clustering methods to effectively identify bi-clusters which can represent some genes only highly related to "partial" specific experimental conditions (or samples) [1]-[5]. For example, Cheng and Church's (CC) algorithm [2] was a pioneering approach to bring bi-clustering into gene expression data analysis. The CC algorithm introduced the mean squared residue measure to identify bi-clusters. The Iterative Signature Algorithm (ISA) [6] first normalized the data points of row and column, respectively, and then chose the greatest Z scores of row and of column simultaneously as the criteria for selection of bi-clusters. The Order Preserving Sub Matrix (OPSM) [7] algorithm used a stochastic model to identify a subset of genes with a coherently relative order among a subset of experimental conditions. The Statistical-Algorithmic Method for Bi-cluster Analysis (SAMBA) [8] first converted a gene expression matrix into a weighted bipartite graph. Then the problem of discovering the most significant bi-clusters was transformed into finding the densest subgraphs in a bipartite graph.

In addition, some bi-clustering approaches have also been successfully applied to text mining. This study applied the fuzzy co-clustering with Ruspini's condition (FCR) method [9] from text mining to extract bi-clusters from microarray data. The FCR method determines the importance of data points in a bi-cluster in row and column dimensions, respectively, by simultaneously using two fuzzy conditions in its objective function to calculate the corresponding weights of each row and each column. Here, we not only applied the FCR method to microarray data analysis, but also proposed an automatic bi-cluster extraction strategy to identify bi-clusters as well. The details are shown in the following sections.

II. Materials and Methods

A. Fuzzy Co-clustering with Ruspini's Condition (FCR)

In clustering, the fuzzy concept is usually adopted to show that a data point can belong to different groups with a different degree of memberships. This fuzzy concept

significantly enhances capability of representation of overlapping clusters. In bi-clustering processes (also called co-clustering processes) especially for text mining (to co-cluster documents and words simultaneously), the fuzzy concept is widely utilized to bring a degree of membership to data representation. In this study, we apply a fuzzy co-clustering concept from text mining [9], called FCR (fuzzy co-clustering with Ruspini's condition), to extraction of bi-clusters (BCs, also called co-clusters) from microarray data. In addition, we design a process to iteratively and efficiently extract each BC. Note that, here FCR is designed to find bi-clusters with consistently high gene expression values for the selected genes under subsets of experimental conditions. The objective function J_R in the FCR algorithm is

$$J_R = \sum_{c=1}^{C}\sum_{i=1}^{N}\sum_{j=1}^{K} u_{ci} v_{cj} x_{ij} + R - T_u \sum_{c=1}^{C}\sum_{i=1}^{N} u_{ci} \ln(u_{ci}) - T_v \sum_{c=1}^{C}\sum_{j=1}^{K} v_{cj} \ln(v_{cj}) \tag{1}$$

where $R = \dfrac{NK - \sum_{c=1}^{C}\sum_{i=1}^{N}\sum_{j=1}^{K} u_{ci} v_{cj}}{NK} \times \sum_{i=1}^{N}\sum_{j=1}^{K} x_{ij}$. The objective function J_R is optimized subject to two constraint $\sum_{c=1}^{C} u_{ci} = 1$ and $\sum_{c=1}^{C} v_{cj} = 1$ for all i and j. The derived formulas for u_{ci} and v_{cj} are shown as follows:

$$u_{ci} = \frac{\exp\left(\dfrac{\sum_{j=1}^{K} v_{cj} x_{ij} - \dfrac{\sum_{j=1}^{K} v_{cj}}{NK} \sum_{p=1}^{N}\sum_{j=1}^{K} x_{pj}}{T_u}\right)}{\sum_{d=1}^{C} \exp\left(\dfrac{\sum_{j=1}^{K} v_{dj} x_{ij} - \dfrac{\sum_{j=1}^{K} v_{dj}}{NK} \sum_{p=1}^{N}\sum_{j=1}^{K} x_{pj}}{T_u}\right)} \tag{2}$$

$$v_{cj} = \frac{\exp\left(\dfrac{\sum_{i=1}^{N} u_{ci} x_{ij} - \dfrac{\sum_{i=1}^{N} u_{ci}}{NK} \sum_{i=1}^{N}\sum_{q=1}^{K} x_{iq}}{T_v}\right)}{\sum_{d=1}^{C} \exp\left(\dfrac{\sum_{i=1}^{N} u_{di} x_{ij} - \dfrac{\sum_{i=1}^{N} u_{di}}{NK} \sum_{i=1}^{N}\sum_{q=1}^{K} x_{iq}}{T_v}\right)} \tag{3}$$

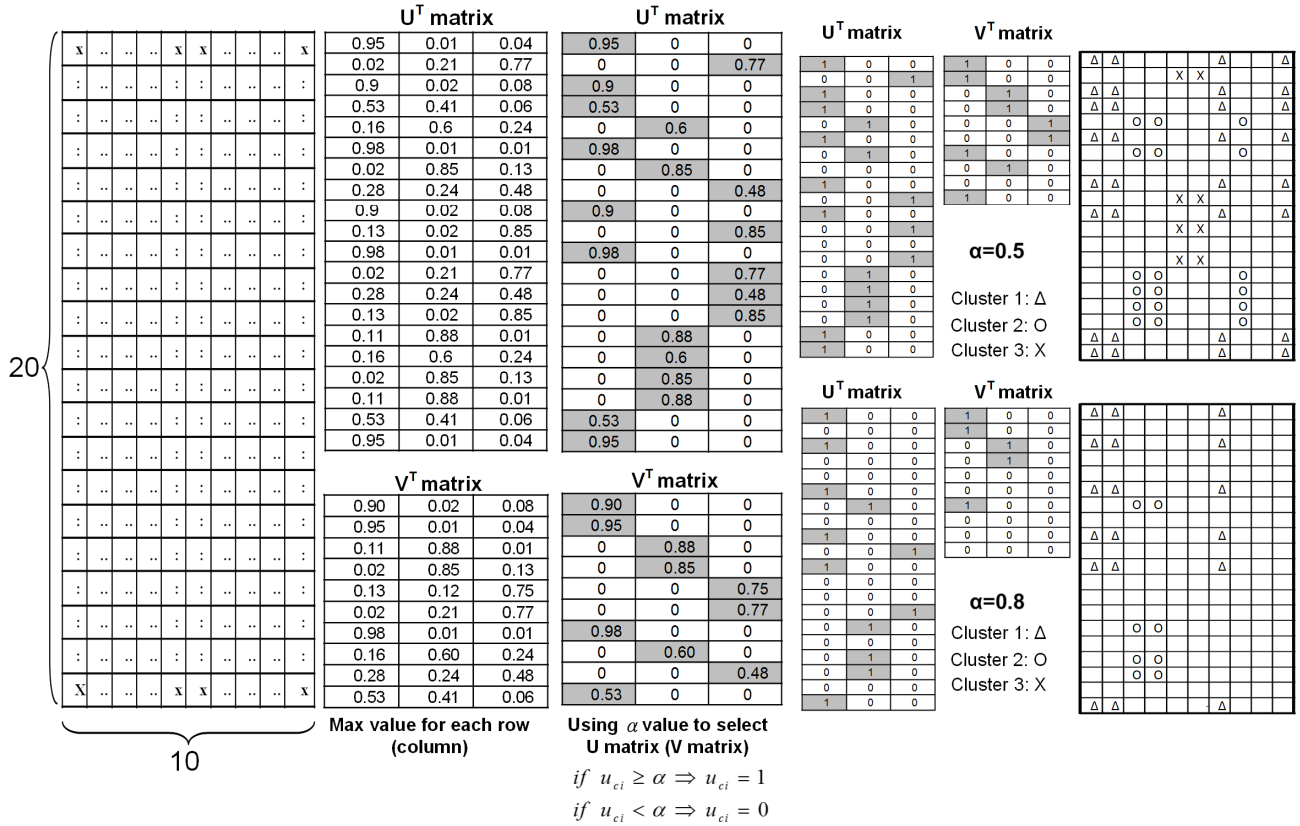

Fig. 1. An illustration of the bi-cluster (BC) determination process.

The values of u_{ci} and v_{cj} shall iteratively update using Eqs. (2) and (3) respectively until convergence is achieved.

The related parameters in (1) for microarray data are explained as follows. N represents the number of genes, K is the number of experimental conditions, C is the predefined number of BCs, and x_{ij} denotes the expression value of the i^{th} gene under the j^{th} experimental condition. In a BC c, u_{ci} and v_{cj} represent the degree of membership for the i^{th} gene and j^{th} experimental condition to cluster c, respectively. T_u and T_v are pre-defined degree of fuzziness parameters. Also note that the main difference between FCR and other fuzzy bi-clustering methods is that FCR creates a degree of membership for both bi-axial variables (e.g., degree of membership for gene axis, u_{ci}, and degree of membership for experimental condition axis, v_{cj}) in order to produce better bi-clustering results, whereas other fuzzy bi-clustering methods only bring a degree of membership to one bi-axial variable.

B. Illustration of Determination of Bi-clusters (BCs)

As mentioned above, after performing the fuzzy co-clustering process (FCR), and assuming we want to find C bi-clusters (BCs) in the data, we get the degree of membership for the i^{th} gene (u_{ci}) and the degree of membership for the j^{th} experimental condition (v_{cj}) to the c^{th} BC. Next we should identify the members (both genes and experiments) of the c^{th} BC based on both u_{ci} and v_{cj} values. An illustration of this complete BC determination process and identification of members in each BC is shown in Fig. 1 and explained below.

In Fig.1, we use a synthetic example with 20 genes and 10 experimental conditions (20×10 data matrix), and seek 3 BCs in the data. Then FCR is used to produce two matrices: U matrix (3×20) represents the degree of membership for 20 genes in 3 BCs and V matrix (3×10) represents the degree of membership for 10 experimental conditions in 3 BCs. As mentioned earlier, the constraints should be satisfied by U and V; i.e., in the U matrix, for each gene the sum of the degrees of membership for 3 BCs (i.e. the sum across each row) should be equal to 1; in the V matrix for each experimental condition the sum of the degrees of membership for 3 BCs (i.e. the sum down each column) should also be equal to 1. Next, for each gene and each experimental condition in the U and V matrices respectively, the degrees of membership for 3 BCs are further processed: the maximum value is kept and the other values are set to 0 (i.e., to keep the strongest degree and discard the weaker degrees of membership for every gene and experimental condition among the specified BCs). From this step, we also obtain numbers of genes and experimental conditions belonging to each BC (e.g., in this case BCs 1~3 have 8, 6, 6 genes, respectively, and 4, 3, 3 experimental conditions, respectively). Finally, we can use a different threshold value, α, to further filter out weaker members for BCs from gene and experimental condition perspectives (i.e., the degree of membership for every gene and experimental condition≥α). For example, in this case when α=0.5, no change happens for BCs 1~2, but the number of genes and experimental conditions in BC 3 have been reduced to 4 and 2, respectively. However, when α=0.8, all three BCs have been influenced (actually, BC 3 is lost because no experimental condition satisfies the constraint). Note that, for simplifying the visualization of the satisfied members for BCs, in Fig. 1 the members of the resulting U and V matrices are set to only two values, 1 and 0, where 1 represents those genes and conditions which meet the α threshold and 0 represents those that do not. From the resulting U and V matrices, because we know the members of genes and experimental conditions for each BC, we can also easily visualize the resulting BCs from the original data matrix, as shown in the blocks furthest to the right in Fig.1.

C. Automatic Bi-cluster (BC) Extraction Strategy

Fig. 2 shows the automatic bi-cluster (BC) extraction process proposed in this study and used on microarray data. The main point of this process is to use FCR to determine the c BCs, and then to further select only one BC to meet the selection criteria that have been set and to remove the selected BC from microarray data. The aforementioned procedure is repeated to extract all BCs existing in the microarray data. The following briefly describes the 2 major functions of this flow diagram.

The first part of Fig. 2 is called preprocessing. It includes the following steps. First, the gene expression values in microarray data are normalized in [0, 1]. Next, the fuzzy c-means approach is used to find $C\times N$ initial values of the matrix U. The update equations for the matrices U and V are derived from (1) and are iteratively used to determine the final values of U and V. Note that, the authors of FCR randomly initialized U. It may make the algorithm require more iterations to converge.

The second part of Fig. 2 is aimed at extraction of the BCs. As mentioned previously, this process finds the BCs one by one in the microarray data. Therefore, after the c BCs have been identified via FCR, we must select one BC according to the following 3 criteria:

1. Determine the number of genes and experimental conditions belonging to each BC: In the c BCs, the u_{ci} and v_{cj} values respectively represent the degree of each row and each column belonging to the c^{th} BC. As shown in Fig. 2, we can set a threshold value α and decide which genes and experimental conditions are worth considering members of a BC (u_{ci} or $v_{cj}\geq\alpha$).

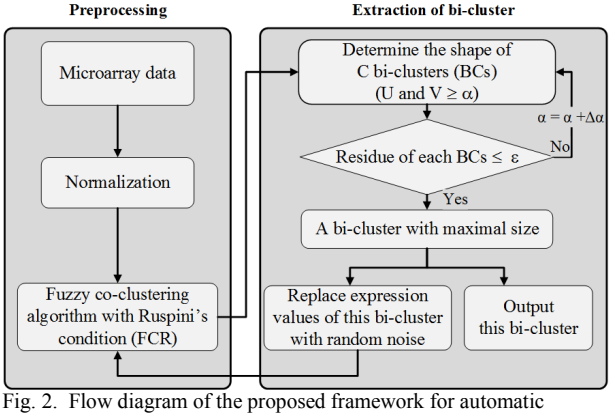

Fig. 2. Flow diagram of the proposed framework for automatic bi-cluster (BC) extraction.

2. The residue constraint of each BC: After deciding the genes and experimental conditions which belong to each BC, whether each BC has consistent high gene expression values or not can be determined via the mean squared residue measure [2]:

$$\text{Residue}(BC) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} \left(x_{ij} - x_{iJ} - x_{Ij} + x_{IJ} \right)^2 \quad (4)$$

Equation (4) shows that I genes and J experimental conditions belong to a certain BC, x_{ij} denotes the expression value of the i^{th} gene under the j^{th} experimental condition, x_{iJ} represents the average expression value of the i^{th} gene among those J experimental conditions, x_{Ij} indicates the average expression value of the j^{th} experimental condition among those I genes, x_{IJ} is the average expression value among those I genes and those J experimental conditions. Here we only select the BCs with the residue values less than the set threshold value ε. However, if the residue values of all BCs exceed ε, then α should be increased to select genes and experimental conditions more strictly and re-examine this condition.

3. Output a BC with the maximum size: After the two aforementioned steps, the size of each BC is determined by calculating the product of the number of associated genes and the number of associated experimental conditions in each BC. Since we want to identify BCs with a bigger size, in this step we only select the BC with maximum size. Every time a BC has been selected, we use random noisy data points to replace the gene expression values in this BC, and then return to the FCR in preprocessing to re-calculate u_{ci} and v_{cj}.

III. Results and Discussion

Here, we demonstrate that our MFCC approach could extract almost every designed bi-cluster (BC) in the synthetic data. Then we further compared our MFCC's performance with four well-known state-of-the-art methods using the synthetic data with different levels of overlapping conditions of designed BCs in both bi-axial variables. Note that, for each data set in the following three cases, the expression values of designed BCs were randomly set to 0.6~1, and the expression values of non-BC regions were randomly set to 0~0.3 to simulate background noise. The values of parameters used in our MFCC approach are shown in Table I. The details of experiments are explained as follows.

TABLE I
THE VALUES OF PARAMETERS USED IN MFCC

Symbols	Values	Notation
C	5	Predefined number of BCs used in (1)
T_u / T_v	1 / 1	Degree of fuzziness parameters used in (1)
α	0.3	Threshold value to determine the members of each BC used in (4)
ε	0.3	Threshold value for residue of each BC used in (4)

A. Case I: Synthetic Data with Simple Overlapping BCs

As shown in Figs. 3(a), 3(e), and 3(i), we have generated three types of synthetic data sets, each with a 100×100 matrix and each with 3 designed BCs in different locations with sizes of 30×30, 30×30, and 15×15, respectively. Here two BCs with the same size (30×30) in each data set separately had an overlapping region in the gene axis (15 overlapping genes, Fig. 3(a)), in experimental condition axis (15 overlapping experimental conditions, Fig. 3(e)), or in two axes (a 10×10 overlapping region, Fig. 3(i)). As shown in Figs. 3(b)-3(d), Figs. 3(f)-3(h), and Figs. 3(j)-3(l), our MFCC approach iteratively and correctly extracted each designed BC in each data set. Note that, in Fig. 3(k), the specific left and upper region showed the extracted BC using a different color (to represent those data points with a low expression value). Actually, the overlapping region for the two BCs in this case was the mentioned specific area. Hence, this overlapping region was filled in with the random noise (with values 0~0.3) after the first BC was extracted by our MFCC approach as mentioned before. In addition, since the size of the specific overlapping region was small enough compared to the size of the second BC, our MFCC approach could extract the complete second BC (30×30).

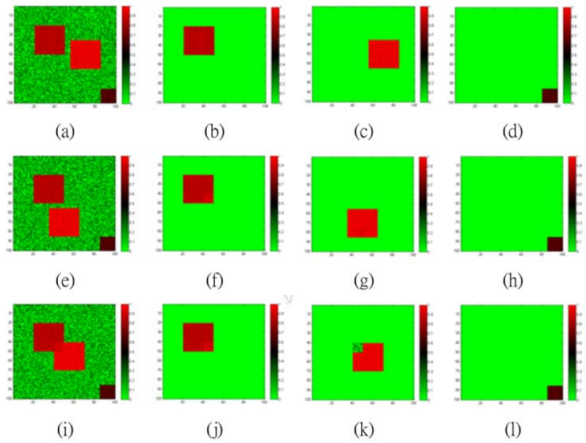

Fig. 3. Synthetic data with 3 BCs and the extraction of BCs by our MFCC approach. Subfigures (a), (e), and (i) are the three synthetic data sets, each with 3 BCs; other subfigures are each corresponding BC interactively extracted by our MFCC approach.

B. Case II: Synthetic Data with More Complicated Overlapping BCs

To further demonstrate the ability to extract BCs by our MFCC approach, we created another synthetic dataset with more complicated overlapping BC conditions. As shown in Fig. 4(a), this data set had a 50×50 matrix and 5 BCs in different locations with sizes of 20×10, 10×20, 10×9, 8×10, and 5×5, respectively. In addition, this data set simultaneously had several overlapping regions among the 5 BCs regarding the gene axis, experimental condition axis, and two axes. The extraction of designed BCs by our MFCC approach are shown in Figs. 4(b)-4(f). Again, our MFCC approach correctly extracted each designed BC in this data set, except for the BC in Fig. 4(d). The reason for the unsuccessful extraction of this BC was similar to the reason mentioned in the previous case. The size of the specific overlapping region (3×3) was not small enough compared to the original size (8×10) of this BC in Fig. 4(d). Hence, our

MFCC approach could only extract a part (6×8) of this BC in Fig. 4(d) regardless of the 2 genes and 2 experimental conditions in the specific overlapping region.

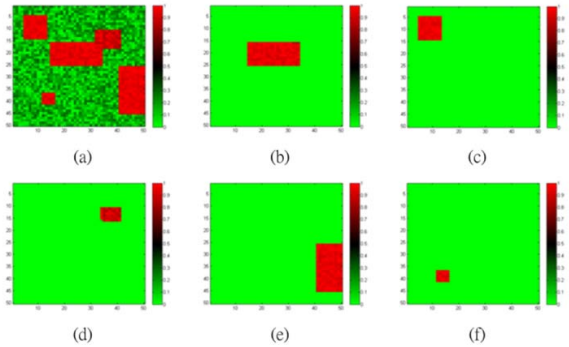

Fig. 4. Synthetic data with 5 BCs and the extraction of BCs by our MFCC approach. Subfigure (a) is the synthetic data sets with 5 BCs; other subfigures are each corresponding BC interactively extracted by our MFCC approach.

C. Case III: Synthetic Data with Different Levels of Overlapping BC Conditions in Both Axes

In this case, we have generated another 10 sets of synthetic data, each with a 100×100 matrix and each with 5 designed BCs. The 5 designed BCs in each data set were put in order along a diagonal, but the size of each overlapping region between 2 adjacent BCs in each data set was changed from 1×1 (denoted overlapping degree = 1, Fig. 5(a)) to 10×10 (denoted overlapping degree = 10, Fig. 5(b)). We used these 10 data sets to perform the comparisons for our MFCC approach and four other well-known bi-clustering methods (SAMBA [8], CC [2], OPSM [7], and ISA [6]). Here we used the BicAT package [10] to execute CC, OPSM, and ISA, and used the EXPANDER package [11] to run SAMBA. In addition, the Prelic's match score [12], [13] was used as a measurement for evaluation. The details of the Prelic's match score was given as follows:

$$S_G(G_1, G_{opt}) = \frac{|G_1 \cap G_{opt}|}{|G_1 \cup G_{opt}|}, \quad S_C(C_1, C_{opt}) = \frac{|C_1 \cap C_{opt}|}{|C_1 \cup C_{opt}|}$$

$$S_G(M_1, M_{opt}) = \frac{\sum_{G_1 \in M_1} \max_{G_{opt} \in M_{opt}} S_G(G_1, G_{opt})}{|M_1|}$$

$$S_C(M_1, M_{opt}) = \frac{\sum_{C_1 \in M_1} \max_{C_{opt} \in M_{opt}} S_C(C_1, C_{opt})}{|M_1|}$$

$$\text{Match Score} = \sqrt{S_G(M_1, M_{opt}) \times S_C(M_1, M_{opt})} \quad (5)$$

where M_{opt} denotes the set of implanted BCs, M_1 represents the set of BCs extracted by a bi-clustering method, G_{opt} and C_{opt} respectively designate the genes and experimental conditions belonging to one of the implanted BCs, G_1 and C_1 respectively indicate the genes and experimental conditions belonging to one of the BCs extracted by a bi-clustering method, $|G_1 \cap G_{opt}|$ and $|G_1 \cup G_{opt}|$ respectively represent the intersection and union of genes between an implanted BC and a BC extracted by a bi-clustering method, $|C_1 \cap C_{opt}|$ and $|C_1 \cup C_{opt}|$ respectively represent the intersection and union of experimental conditions between an implanted BC and a BC extracted by a bi-clustering method. Furthermore, $S_G(M_1, M_{opt})$ and $S_C(M_1, M_{opt})$ in (5) measure the extent to which the BCs extracted by a bi-clustering method represent the implanted BCs in the gene axis and experimental condition axis, respectively. The Prelic's match score is defined as the geometric mean of $S_G(M_1, M_{opt})$ and $S_C(M_1, M_{opt})$.

Results of the comparison of the tools were shown in Fig. 6. Here we have observed some interesting phenomena: (1) Our MFCC approach extracted almost all designed BCs regardless of the size of overlapping regions between 2 adjacent BCs. The reason is the same as that with case 1: the size of the specific overlapping region between two BCs was small enough compared to the size of the second extracted BC. (2) Some bi-clustering methods, such as CC, OPSM, and ISA, had a quite low Prelic's match score. It may be because we used default values for parameters suggested for these methods. (3) SAMBA suffers the most severely when increasing the overlapping degree followed in order by ISA and CC, respectively.

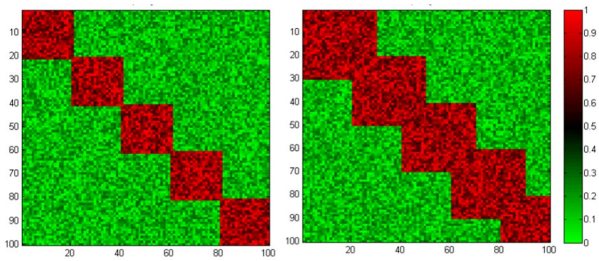

Fig. 5. Synthetic data with different levels of overlapping BC conditions in both axes. (a) Each of overlapping regions with 1 gene and 1 experimental condition; (b) Each of overlapping regions with 10 genes and 10 experimental conditions.

IV. CONCLUSION

In order to effectively extract bi-clusters from microarray data, this research proposed a modified fuzzy co-clustering (MFCC) approach. This approach was applied to several synthetic data with various degrees of overlapping conditions. The results proved the effectiveness of the proposed approach. In addition, when considering the synthetic data with different levels of designed overlapping bi-cluster conditions in both bi-axial variables, we have further demonstrated that our MFCC approach can obtain a better Prelic's match score and is not influenced by the different levels of overlapping regions of bi-clusters compared with the four well-known state-of-the-art methods.

In the future, we will continue to modify and verify this approach. There are two aspects that can be further developed:

(1) Although we have demonstrated that our proposed MFCC approach can extract almost all designed each bi-cluster in the synthetic data, we plan to apply this approach to some real microarray data and explore the biological meanings of the extracted bi-clusters. (2) We did not discuss the influence for setting the parameters used in this approach. In the future, feasible methods will be further explored for tuning the parameters to let the proposed system achieve a better performance in identification of bi-clusters.

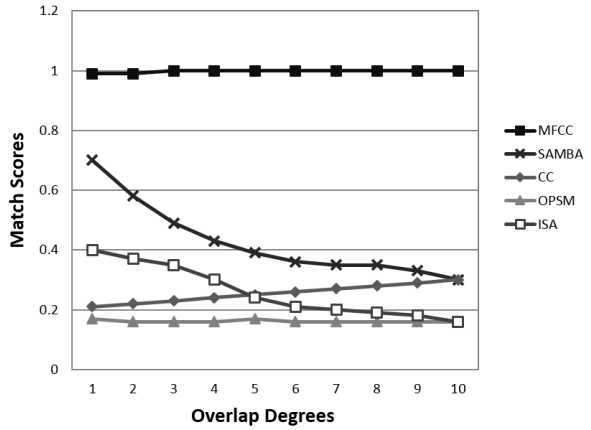

Fig. 6. Performance comparison for our MFCC approach and other 4 bi-clustering methods regarding 10 sets of synthetic data under different levels of overlapping BC conditions in both gene and experimental condition axes.

V. Acknowledgment

We specially thank Prof. Nikhil R. Pal of the Indian Statistical Institute for his great help, suggestions and comments for this study.

References

[1] G. Getz, E. Levine, and E. Domany, "Coupled two-way clustering analysis of gene microarray data," *Proc Natl Acad Sci U S A,* vol. 97, no. 22, pp. 12079–12084, Oct. 2000.

[2] Y. Cheng, and G. M. Church, "Biclustering of expression data," *Proc Int Conf Intell Syst Mol Biol,* vol. 8, pp. 93–103, 2000.

[3] J. Yang, H. Wang, W. Wang, and P. Yu, "Enhanced biclustering on expression data." *Third IEEE Symposium on Bioinformatics and Bioengineering*, pp. 321–327, Mar. 2003.

[4] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein, "Spectral biclustering of microarray data: coclustering genes and conditions," *Genome Res,* vol. 13, no. 4, pp. 703–716, Apr. 2003.

[5] S. C. Madeira, and A. L. Oliveira, "Biclustering algorithms for biological data analysis: a survey," *IEEE/ACM Trans Comput Biol Bioinform,* vol. 1, no. 1, pp. 24–45, Jan.-Mar. 2004.

[6] S. Bergmann, J. Ihmels, and N. Barkai, "Iterative signature algorithm for the analysis of large-scale gene expression data," *Phys Rev E Stat Nonlin Soft Matter Phys,* vol. 67, no. 3 Pt 1, pp. 031902, Mar, 2003.

[7] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini, "Discovering local structure in gene expression data: the order-preserving submatrix problem," *J Comput Biol,* vol. 10, no. 3–4, pp. 373–84, Jun. 2003.

[8] A. Tanay, R. Sharan, and R. Shamir, "Discovering statistically significant biclusters in gene expression data," *Bioinformatics,* vol. 18, no. suppl 1, pp. S136–S144, Mar. 2002.

[9] W. C. Tjhi, and L. Chen, "A partitioning based algorithm to fuzzy co-cluster documents and words," *Pattern Recognition Letters,* vol. 27, no. 3, pp. 151–159, Feb. 2006.

[10] S. Barkow, S. Bleuler, A. Prelić, P. Zimmermann, and E. Zitzler, "BicAT: a biclustering analysis toolbox," *Bioinformatics,* vol. 22, no. 10, pp. 1282–1283, Mar. 2006.

[11] R. Sharan, A. Maron-Katz, and R. Shamir, "CLICK and EXPANDER: a system for clustering and visualizing gene expression data," *Bioinformatics,* vol. 19, no. 14, pp. 1787–1799, Sep. 2003.

[12] A. Preli, 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, Feb. 2006.

[13] K. Eren, M. Deveci, O. Küçüktunç, and Ü. V. Çatalyürek, "A comparative analysis of biclustering algorithms for gene expression data," *Briefings in Bioinformatics*, vol. 14, no. 3, pp. 279–92, Apr. 2012.