Linear Coherent Bi-cluster Discovery via Line Detection and Sample Majority Voting

Yi Shi, Zhipeng Cai, Guohui Lin, Dale Schuurmans

Department of Computing Science
University of Alberta, Edmonton, Alberta T6G 2E8
{ys3,zhipeng,ghlin,dale}@cs.ualberta.ca

Abstract. Discovering groups of genes that share common expression profiles is an important problem in DNA microarray analysis. Unfortunately, standard bi-clustering algorithms often fail to retrieve common expression groups because (1) genes only exhibit similar behaviors over a subset of conditions, and (2) genes may participate in more than one functional process and therefore belong to multiple groups. Many algorithms have been proposed to address these problems in the past decade; however, in addition to the above challenges most such algorithms are unable to discover linear coherent bi-clusters—a strict generalization of additive and multiplicative bi-clustering models. In this paper, we propose a novel bi-clustering algorithm that discovers linear coherent bi-clusters, based on first detecting linear correlations between pairs of gene expression profiles, then identifying groups by sample majority voting. Our experimental results on both synthetic and two real datasets, *Saccharomyces cerevisiae* and *Arabidopsis thaliana*, show significant performance improvements over previous methods. One intriguing aspect of our approach is that it can easily be extended to identify bi-clusters of more complex gene-gene correlations.

1 Introduction

Microarray analysis involves monitoring the expression levels of thousands of genes simultaneously over different conditions. Although such an emerging technology enables the language of biology to be spoken in mathematical terms, extracting useful information from the large volume of experimental microarray data remains a difficult challenge. One important problem in microarray analysis is to identify a subset of genes that have similar expression patterns under a common subset of conditions. Standard clustering methods, such as K-means clustering [8, 5], hierarchical clustering [16, 19] and self-organizing map [17], are usually not suitable for microarray data analysis for two main reasons: (1) genes exhibit similar behaviors not over all conditions, but over a subset of conditions, and (2) genes may participate in more than one functional processes and hence belong to multiple groups. Thus, traditional clustering algorithms typically do not produce a satisfactory solution. To overcome the limitations of the traditional clustering methods, the concept of bi-clustering was developed where one

seeks groups of genes that exhibit similar expression patterns, but only over a subset of the sample conditions. Figure 1 illustrates a gene expression matrix without any obvious bi-clusters (left) and an expression matrix with a salient bi-cluster (right).

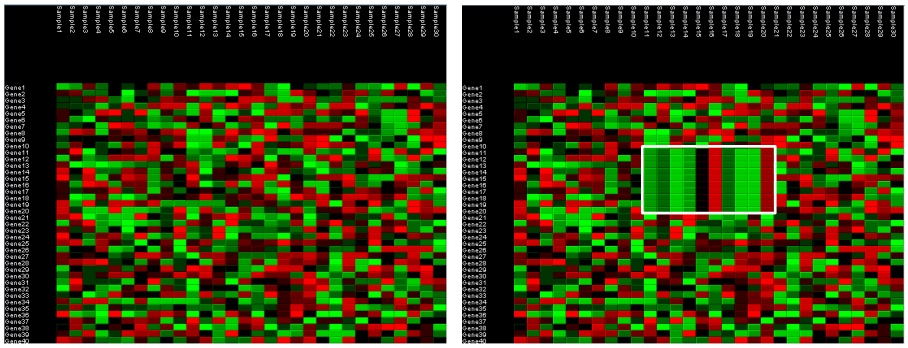

Fig. 1. Example of a constant row bi-cluster in gene expression matrix. The left image shows a gene expression matrix without any obvious bi-clusters; the right image shows an expression matrix with a constant row bi-cluster.

The term bi-clustering, also called co-clustering, or two-mode clustering was first mentioned by Hartigan in [7] and latter formalized by Mirkin in [14]. Cheng and Church [4] were the first to apply bi-clustering to gene expression analysis. Since then, dozens of bi-clustering algorithms have been proposed for the gene expression analysis. The general bi-clustering problem and many of its variants were proved to be NP-hard in [4], and therefore most bi-clustering algorithms comprise heuristic approaches unless special restrictions are made on the bi-cluster type and(or) bi-cluster structure. Among such bi-clustering algorithms, the majority assume that a expression matrix contains multiple bi-clusters rather than a single bi-cluster. Under the multiple bi-cluster circumstance, different bi-cluster structures can be considered, such as exclusive row and(or) column bi-clusters, checkerboard structure bi-clusters, non-overlapping tree-structured bi-clusters, non-overlapping non-exclusive bi-clusters, overlapping bi-clusters with hierarchical structure, arbitrarily positioned overlapping bi-clusters, and arbitrarily positioned overlapping bi-clusters [13]. The specific variant of the problem we address with the algorithm proposed in this paper, the Linear Coherent Bi-cluster Discovering (LCBD) algorithm, is the last form of bi-cluster structure; i.e., arbitrarily positioned overlapping bi-clusters. This last form is a a more general structure that covers most of the other bi-cluster structures.

Before designing a bi-clustering algorithm, one needs to determine what type (model) of individual bi-clusters to be looking for. There are six primary types considered in the literature, illustrated in Figure 2: (a) the constant value model, (b) the constant row model, (c) the constant column model, (d) the additive co-

herent model, where each row or column is obtained by adding a constant to another row or column, (e) the multiplicative coherent model, where each row or column is obtained by multiplying another row or column by a constant value, and (f) the linear coherent model, where each column is obtained by multiplying another column by a constant value and then adding a constant [6]. To understand which type of bi-cluster structure makes the sense for gene expression analysis, one should note that the ultimate purpose is to identify pairs of biologically related genes such that, under certain conditions, one activates or deactivates the other, either directly or indirectly, during a genetic regulatory process. Because a gene may regulate a group of other genes, this problem becomes identifying groups of such genes, i.e., bi-clusters. Housekeeping genes, which are constitutively expressed over most conditions, are not biologically or clinically interesting. The genes that the first two bi-cluster models, i.e., (a) and (b) find tend to be this kind. Therefore, most existing algorithms are based on either the additive model (d) or the multiplicative model (e) [6]. Since type (f) is a more general type that unifies types (c), (d), and (e), we focus on seeking type (f) bi-clusters in this paper.

x	y	z	w
1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0

(a)

x	y	z	w
1.2	1.2	1.2	1.2
0.8	0.8	0.8	0.8
1.5	1.5	1.5	1.5
0.6	0.6	0.6	0.6

(b)

x	y	z	w
1.2	0.8	1.5	0.6
1.2	0.8	1.5	0.6
1.2	0.8	1.5	0.6
1.2	0.8	1.5	0.6

(c)

x	y	z	w
1.2	0.8	1.5	0.6
1.0	0.6	1.3	0.4
2.0	1.6	2.3	1.4
0.7	0.3	1.2	0.3

(d)

x	y	z	w
2.0	4.0	8.0	1.0
1.0	2.0	4.0	0.5
4.0	8.0	16.0	2.0
1.0	2.0	4.0	0.5

(e)

x	y	z	w
2.0	4.0	3.0	5.0
1.5	2.5	2.0	3.0
2.3	4.3	3.3	5.3
4.5	8.5	6.5	10.5

(f)

Fig. 2. Examples of different bi-cluster types: (a) constant value model; (b) constant row model; (c) constant column model; (d) additive coherent model; (e) multiplicative coherent model; (f) linear coherent model

Our algorithm, the Linear Coherent Bi-cluster Discovering (LCBD) algorithm, is based on first detecting linear correlations between pairs of gene expression profiles, then identifying groups by sample majority voting. To evaluate our algorithm, we will compare its performance to six existing, well known biclustering algorithms: Cheng and Church's algorithm, CC [4]; Samba [18]; Or-

der Preserving Sub-matrix Algorithm, OPSM [1]; Iterative Signature Algorithm, ISA [10, 9]; Bimax [15]; and Maximum Similarity Bi-clusters of Gene Expression Data, MSBE [12]. The first five algorithms were selected and implemented in the survey [15]. The last algorithm, MSBE, is the first polynomial time bi-clustering algorithm that finds optimal solutions, but under certain constraints. To briefly explain each of the first five bi-clustering algorithms: in [4] Cheng and Church defined a merit score, called mean squared residue, to evaluate the quality of a bi-clustering, and then develop a greedy algorithm for finding δ-bi-clusters. Yang *et al.* improved Cheng and Church's method by allowing missing values in gene expression matrices. Tanay *et al.* [18] and Prelić *et al.* [15] search for bi-clusters of up-regulated or down-regulated expression values, while the original expression matrices are discretized to binary matrices during a pre-processing phase. Ihmels *et al.* [10, 9] used gene and condition signatures to evaluate bi-clusters, and propose a random iterative signature algorithm (ISA) when no prior information of the matrix is available. Ben-Dor *et al.*[1] attempt to find the order-preserving sub-matrix (OPSM) bi-clusters in which all genes have same linear ordering, based on a heuristic algorithm.

The remainder of the paper is organized as follows: First, we present the details of our LCBD method in Section 2. Then Section 3 describes the experimental evaluation of our proposed method, comparing its performance on both synthetic and real data to other algorithms. Section 4 then assesses the advantages and disadvantages of the LCBD algorithm and proposes some possible approaches that may overcome the drawbacks of the LCBD algorithm.

2 Methods and Algorithms

Let $A(I, J)$ be an $n \times m$ real valued matrix, where $I = \{1, 2, 3, ..., n\}$ is the set of genes and $J = \{1, 2, 3, ..., m\}$ is the set of samples. The element a_{ij} of $A(I, J)$ represents the expression level of gene i under sample j. A row vector $A(i, J)$ and a column vector $A(I, j)$ represents the ith gene over all the samples and the jth sample over all the genes, respectively. Our algorithm is composed of three major steps.

In the first step, for each pair of genes $A(p, J)$ and $A(q, J)$, where $p, q \in \{1, 2, 3, ...n\}$ and $p \neq q$, we construct a two-dimension binary matrix that represents the 2D image of the two vectors with x-coordinates $A(p, J)$ and y-coordinates $A(q, J)$, respectively. A pixel in the 2D image is denoted by a 1 in the binary matrix. Using the binary matrix as input, we then identify lines in the 2D image based on the Hough transform. The Hough transform technique works on the following principle: First note that each point (pixel) in a 2D image can be passed through by an infinite number of lines, and each of which can be parameterized by r and θ, where r is the perpendicular distance between the origin and the line and θ is the angle between the perpendicular line and the x-coordinate. Then note that the set of lines that pass through a point forms a sinusoidal curve in the $r - \theta$ coordinate space. Now, if there is a common line that passes through a set of points in the original 2D image, their corresponding

sinusoidal curves must have a point of intersection in the $r - \theta$ space. So by finding a point of intersection in $r - \theta$ space, one can identify a line that passes through a set of points in the original 2D space. Each line in the 2D image is a linear correlation between a pair of genes under a subset of samples. To allow for possible overlaps in the final bi-clusters, we let the Hough transform identify at most k non-reduplicative lines. Therefore, for each pair of genes, we can collect at most k sample sets over which the two genes are linearly correlated. After collecting sample sets for each gene pair, we obtain an $n \times n$ upper trianglular matrix, where each element contains at most k sample sets (See Figure 3 for an illustration). We denote each element in the matrix as S_{ij}. Note that for each 2D image, the horizontal lines and vertical lines in the 2D image are not eliminated since they might not represent linear correlations.

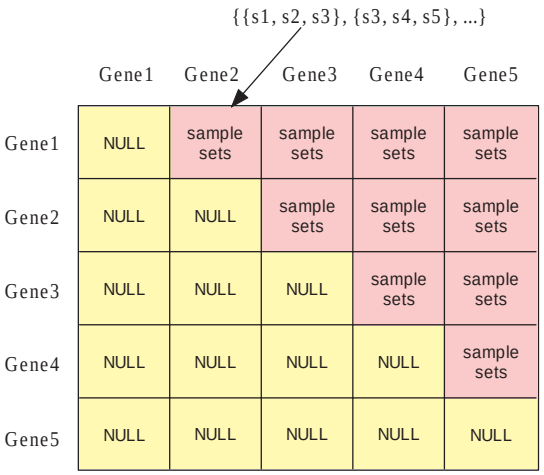

Fig. 3. Illustration of an $n \times n$ gene pairwise sample sets matrix

In the second step, for vector of sample sets, S_{iJ}, we count the samples that appear in each element of S_{iJ}. We then collect the top w voted samples into a sample pool and the corresponding genes who voted for these samples into a gene pool. The sample and gene pools thus constitute an initial bi-cluster. Then, for the remaining samples, we iteratively add them and their corresponding gene into the sample and gene pools, respectively, as long as by adding them the mean gene-gene correlation coefficient of the current bi-cluster remains above a threshold. The user specified parameter w should be greater than 3, because one can always draw a line between any 2 random points but the possibility that more than 2 random points lie on the same line is very small unless there is a linear correlation. In this step, each sample sets vector S_{iJ} will construct at most one bi-cluster that necessarily contains gene i.

In the third step, we remove redundancy in the bi-cluster sets generated in step two. If two bi-clusters share more than 60% identical elements, one of the two will be removed depending on which has more identical elements. Algorithm 1 describes the LCBD algorithm.

Algorithm 1 The LCBD Algorithm

Input An $n \times m$ real value matrix $A(I, J)$, k, w.
Output A set of bi-clusters $A(g_i, s_i)$, where $g_i \subseteq I$ and $s_i \subseteq J$.

for $i = 1$ to n **do**
  for $j = i + 1$ to n **do**
    Construct binary matrix $B_{i,j}$ for vectors $A(i, J)$ and $A(j, J)$;
    Do Hough transform based on $B_{i,j}$ and k to obtain a set of sample sets, S_{iJ};
  end for
end for

for $i = 1$ to n **do**
  Select the top w most voted samples in S_{iJ} as the initial sample pool s_i;
  Select the genes whose corresponding gene pair sample sets contain all the initial samples g_i;
  Construct the initial bi-cluster $A(g_i, s_i)$;
  while gene-wise mean correlation coefficient of $A(g_i, s_i) <$ threshold **do**
    Add the most voted sample in the leftover sample sets to the sample pool s_i;
    Add the corresponding gene into the gene pool g_i;
    Update bi-cluster $A(g_i, s_i)$;
  end while
end for

Remove redundant bi-clusters in the set $A(g_i, s_i)$ that has $> 60\%$ overlapping elements.
Output the set of bi-clusters $A(g_i, s_i)$;

3 Results

We tested our algorithm on both synthetic datasets and two real datasets *Saccharomyces cerevisiae* and *Arabidopsis thaliana*. For the synthetic datasets, we evaluate the algorithms based on how well they identify the real bi-clusters embedded in the expression matrix beforehand. We adopt the Prelić's match score function [15] as a quantified evaluation of merit: Let M_1, M_2 be two sets of bi-clusters. The gene match score of M_1 with respect to M_2 is given by the function

$$S_G^*(M1, M2) = \frac{1}{M_1} \sum_{(G_1,C_1)\in M_1} max_{(G_2,C_2)\in M_2} \frac{|G_1 \cap G_2|}{|G_1 \cup G2|}$$

$S^*_G(M1, M2)$ reflects the average of the maximum match scores for all bi-clusters in M_1 with respect to the bi-clusters in M_2. In our experiment, M_2 is one or more reference (optimal) bi-cluster(s) embedded in the expression matrix beforehand. For the parameter settings of the existing algorithms, we follow the previous works [12] and [15].

3.1 Results on Synthetic Data

Because most existing bi-clustering algorithms do not work on linear coherent bi-clusters, we select two bi-clustering algorithms OPSM and ISA that seek additive bi-cluster structures to compare to our LCBD algorithm, since an additive bi-cluster is a special case of a linear coherent bi-cluster. For the MSBE algorithm, we found in our testing that prior knowledge of a reference gene and reference sample for recovering a synthetic bi-cluster had a great effect on its final result, we therefore do not include the MSBE algorithm into the synthetic experiments because we assume that this prior knowledge is blind to all the algorithms tested.

Constant bi-cluster To produce an expression matrix with an additive bi-cluster, we first randomly generated an 100×50 matrix. The values of the expression matrix obey either a normal distribution (with mean 0 ad SD 1) or a unique distribution (with minimum 0 and maximum 1), since a real data distribution could be either one of them [3, 11, 6]. Within the expression matrix, we randomly select a row and 10 columns to form a size 10 reference gene vector. We then randomly select 9 other row vectors under the same samples and re-calculate their expression values based on the equation $A(i, J_r) = m_i \times A(i_0, J_r) + b_i$, where $A(i_0, J_r)$ is the reference gene vector, m_i equals to 1, and b_i is a random constant. Random noise is then added to the synthetic bi-cluster: a certain percent of elements in the bi-cluster is randomly selected and replaced with random values which obey the same distribution as the background matrix. We tested noise levels of 0% to 25% with increasing steps of 5%. At each noise level we generated 50 synthetic matrices with bi-clusters and reported a final match score that is the mean over the 50 results. Figure 4 shows that our LCBD algorithm obtained the highest match scores for all noise levels and distributions, compared to the two additive bi-cluster type algorithms OPSM and ISA. As one can see, the LCBD algorithm is robust to noise even at noise level 25%. This occurs a line will be identified by the Hough transform as long as it passes through at least 3 points (samples) and during the majority sample voting. Although the expression value under some samples is destroyed, there are sufficiently many others that their expression values under these samples are not destroyed and thus these samples still obtain more votes than random samples that are not within the linear coherence bi-cluster.

Linear coherent bi-cluster Because the LCBD algorithm seeks bi-clusters of the linear coherent type, we can then test it directly on the linear coherent bi-clusters. In this experiment, we only use unique distribution expression

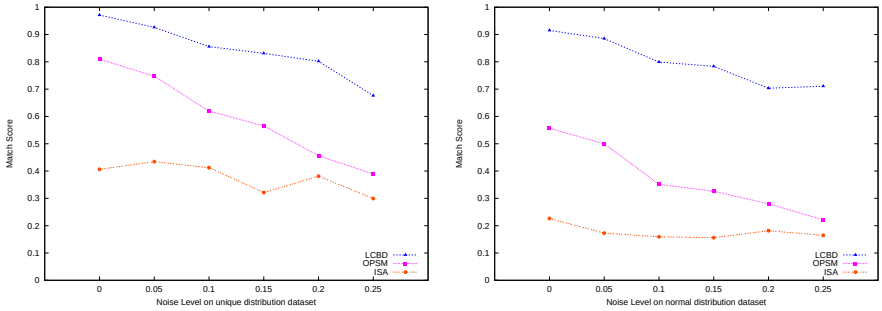

Fig. 4. Match scores of different additive model bi-clustering algorithms on synthetic dataset under unique distribution and normal distribution

matrix, since the normal distribution matrix shows similar results. To generate a linear coherent bi-cluster in a expression matrix, we use the same procedure as in the constant bi-cluster experiment, except that in the equation $A(i, J_r) = m_i \times A(i_0, J_r) + b_i$, the m_i's are no longer 1's but random values. The left part of Figure 5 shows the match scores of the LCBD algorithm under different noise levels and different bi-cluster sizes; the right part of Figure 5 shows the corresponding gene discovery rates of the LCBD algorithm under the same noise level and bi-cluster size. From Figure 5, we can see that the match score and gene discovery rates are generally higher on larger bi-clusters. This is the case because whether a line can be identified during the Hough transform depends more on the absolute number of points that a line passes through than the proportion of points a line passes through. This suggests that the LCBD algorithm should be better at discovering large bi-clusters.

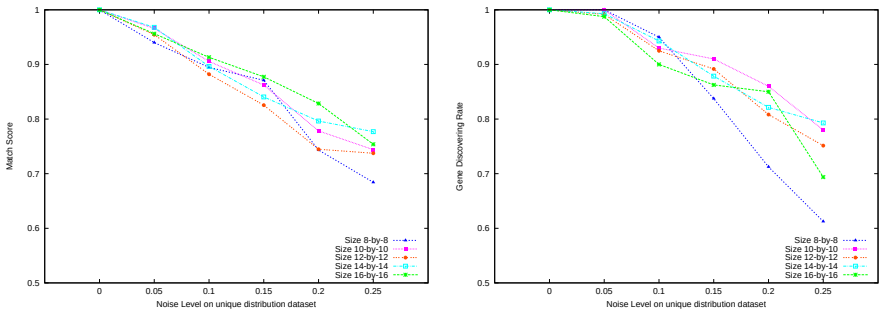

Fig. 5. Match score and gene discovering rate of the LCBD method on synthetic dataset of different bi-cluster size and noise level under unique distribution

Overlapping test To test the LCBD algorithm on discovering multiple overlapping bi-clusters, we generated two linear coherent bi-clusters in the expression matrix and let them overlap to some degree. Figure 6 shows the mean match scores of the LCBD algorithm on discovering two overlapping bi-clusters at noise level 10%. For the overlapping elements, we replace their original values with the sum of the two overlapping values. The overlapping elements are not linear coherent elements and can be viewed as noise elements.

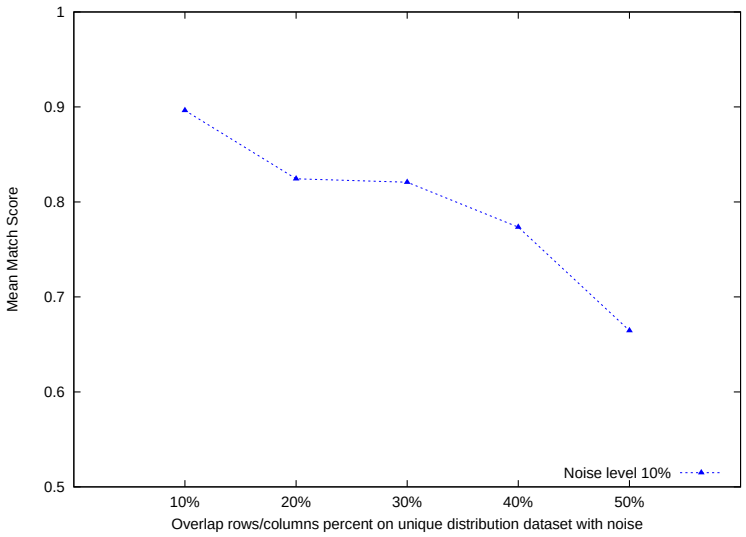

Fig. 6. Match score of the LCBD method of different bi-cluster overlapping rate on synthetic dataset under unique distribution

3.2 Results on Real Data

The documented descriptions of functions and processes that genes participate in has become widely available prior knowledge. The Gene Ontology Consortium in particular provides one of the largest organized collection of gene annotations. Following the idea in [18, 15], we investigate whether the genes identified in bi-clusters produced by the different algorithms show significant enrichment with respect to a specific Gene Ontology annotation. We use two web-servers, FuncAssociate [2] and EasyGo [21], to evaluate the groups of genes produced in our bi-clustering results. The FuncAssociate computes the hypergeometric functional enrichment score, cf. [2], based on Molecular Function and Biological Process annotations. The resulting scores are adjusted for multiple testing by using the Westfall and Young procedure [20, 2]. The EasyGo calculates the functional enrichment score in a similar way. In detail, based on availability, we

tested the bi-clustering results from the *Saccharomyces Cerevisiae* dataset on the FuncAssociate web-server and the results from the *Arabidopsis thaliana* dataset on the EasyGo web-server. The *Saccharomyces Cerevisiae* dataset contains 2993 genes and 173 conditions and the *Arabidopsis thaliana* dataset contains 734 genes and 69 conditions. Figure 7 and Figure 8 show the proportion of gene groups of bi-clusters that are functionally enriched at different significance levels. The LCBD algorithm demonstrates the best results (all 100%) on the *Arabidopsis thaliana* dataset, compared to other seven algorithms; The LCBD results are also competitive to the best results derived from the MSBE algorithm on the *Saccharomyces Cerevisiae* dataset. These results on real datasets indicate that linear coherent bi-clusters are a useful form of bi-cluster structure to extract from gene expression datasets, and could be a bi-cluster type that exists widely in other gene expression datasets.

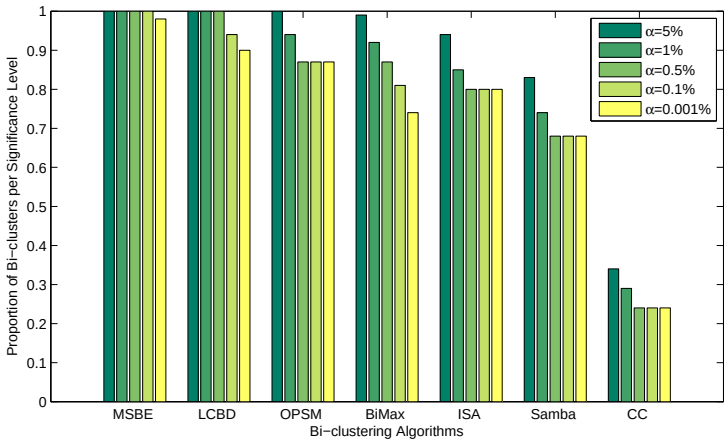

Fig. 7. Proportion of bi-clusters significantly enriched by any GO biological process category (S. Cerevisiae). α is the adjusted significant scores of the bi-clusters

4 Discussion and conclusion

In this paper, we have developed a novel bi-clustering algorithm, the Linear Coherent Bi-cluster Discovering algorithm (LCBD), which seeks linear coherent bi-clusters in gene expression data. Our experimental results on the synthetic data show that the LCBD algorithm can accurately discover additive and linear coherent bi-clusters, while being robust to the noise level and bi-cluster size. Our results on the two real datasets revealed that the linear coherent bi-clusters

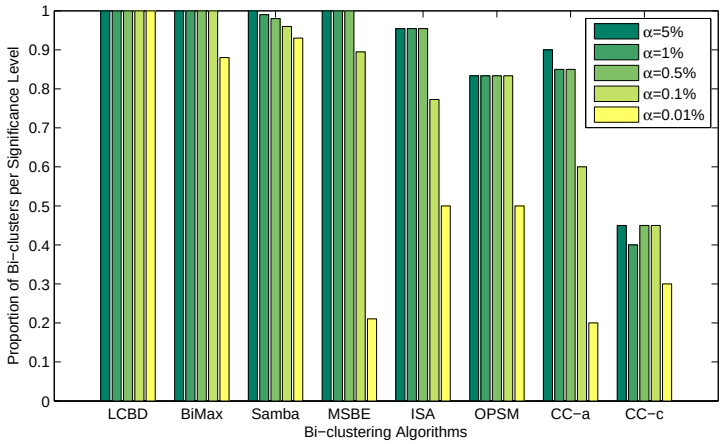

Fig. 8. Proportion of bi-clusters significantly enriched by any GO biological process category (A. thaliana). α is the adjusted significant scores of the bi-clusters

discovered by LCBD are functionally enriched and therefore biologically meaningful.

The drawback of using the traditional Hough transform technique for identifying linear correlations is that it can suffer from sparse data problems: even if some points lie perfectly on a common line, if the binary pixel matrix is too sparse, the traditional Hough transform might not find this line because different parameter values are needed to make the transform work appropriately for different sparsity levels. The sparse data problem may occur when the sample size of the expression matrix is small. However, the sparse data problem can be addressed by applying more advanced image analysis techniques such as the sparse resistant Hough transform or other feature recognition techniques.

The time complexity of the LCBD algorithm is worse than most algorithms mentioned in this paper, and improving its efficiency is an important direction for future work. It appears that selecting a set of representative genes, rather than all genes, to construct the $n \times n$ sample set matrix is a promising approach, since redundancies often occur if ones uses all genes. One intriguing aspect of the LCBD algorithm is that it can easily be extended to identify bi-clusters of more complex gene-gene correlations.

References

1. A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini. Discovering local structure in gene expression data: The order-preserving sub-matrix problem. In *Proc. of the 6th Annual International Conference on Computational Biology*, pages 49–57, 2002.

2. G. F. Berriz, O. D. King, B. Bryant, C. Sander, and F. P. Roth. Characterizing gene sets with funcassociate. *BioInformatics*, 19:2502–2504, 2003.
3. H. C. Causton, J. Quackenbush, and A. Brazma. Microarray gene expression data analysis : A beginner's guide. *Blackwell Publishing, Malden*, 2003.
4. Y. Cheng and G. M. Church. Biclustering of expression data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 93–103, 2000.
5. M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*, 95:14863–14868, 1998.
6. X. Gan, A. W.-C. Liew, and H. Yan. Discovering biclusters in gene expression data based on high-dimensional linear geometries. *BMC Bioinformatics*, 9:209, 2008.
7. J. A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67:123–129, 1972.
8. J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Applied Statistics*, 28:100–108, 1979.
9. J. Ihmels, S. Bergmann, and N. Barkai. Defining transcription modules using large scale gene expression data. *Bioinformatics*, 20:1993–2003, 2004.
10. J. Ihmels, G. Friedlander, S. Bergmann, O. Sarig, Y. Ziv, and N. Barkai. Revealing modular organization in the yeast transcriptional network. *Nature Genetics*, 31:370–377, 2002.
11. Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein. Spectral biclustering of microarray data: coclustering genes and conditions. *Genome Res.*, 13:703–716, 2003.
12. X. Liu and L. Wang. Computing the maximum similarity bi-clusters of gene expression data. *Bioinformatics*, 23:50–56, 2006.
13. S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *Computational Biology and Bioinformatics*, 1:24–45, 2004.
14. Mirkin and Boris. Mathematical classification and clustering. *Kluwer Academic Publishers*, 1996.
15. A. Prelić, S. Bleuler, P. Zimmermann, and A. Wille. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, 22(9):1122–1129, 2006.
16. R. R. Sokal and C. D. Michener. A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, 38:1409–1438, 1958.
17. P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, and T. R. Golub. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proceedings of the National Academy of Sciences of the United States of America*, 96:2907–2912, 1999.
18. A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18:136S–144, 2002.
19. S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church. Systematic determination of genetic network architecture. *Nature Genetics*, 22:281–285, 1999.
20. P. H. Westfall and S. S. Young. Resampling-based multiple testing. *Wiley, New York*, 1993.
21. Z. S. X. Zhou. Easygo: Gene ontology-based annotation and functional enrichment analysis tool for agronomical species. *BMC Genomics*, 8:246, 2007.