Linear coherent bi-cluster discovery via beam detection and sample set clustering

Yi Shi [*], Maryam Hasan, Zhipeng Cai, Guohui Lin, and Dale Schuurmans

Department of Computing Science, University of Alberta
Edmonton, Alberta T6G 2E8, Canada
{ys3,mhasan1,zhipeng,ghlin,dale}@cs.ualberta.ca

Abstract. We propose a new bi-clustering algorithm, LinCoh, for finding *linear coherent* bi-clusters in gene expression microarray data. Our method exploits a robust technique for identifying conditionally correlated genes, combined with an efficient density based search for clustering sample sets. Experimental results on both synthetic and real datasets demonstrated that LinCoh consistently finds more accurate and higher quality bi-clusters than existing bi-clustering algorithms.

Keywords: Bi-clustering, sample set clustering, gene expression microarray, gene ontology.

1 Introduction

Gene expression microarray data analysis interprets the expression levels of thousands of genes across multiple conditions (also called samples). Such a study enables the language of biology to be spoken in mathematical terms; however, it remains a challenge to extract useful information from the large volume of raw expression data.

One central problem in gene expression microarray data analysis is to identify groups of genes that have similar expression patterns in a common subset of conditions. Standard clustering methods, such as k-means clustering [4], hierarchical clustering [21] and self-organizing maps [20], are ill-suited to this purpose for two main reasons: that genes exhibit similar behaviors only under some, but not all conditions, and that genes may participate in more than one functional process and hence belong to multiple groups. Bi-clustering [9,16] is intended to overcome the limitations of standard clustering methods by identifying a group of genes that exhibit similar expression patterns in a subset of conditions. Bi-clustering was first applied to gene expression analysis a decade ago [3], subsequently leading to dozens of other bi-clustering algorithms. Nevertheless, the general bi-clustering problem is NP-hard [3]. Efforts were invested in designing bi-clustering algorithms, mostly heuristics, for finding postulated types of bi-clusters.

[*] Correspondence author. Tel: +1 (780) 492 2285 ; Fax: +1 (780) 492 6393.

There are several types of bi-clusters that have been sought previously, including (a) the constant value model, (b) the constant row model, (c) the constant column model, (d) the additive coherent model, where each row (or column) is obtained by adding a constant to another row (or column, respectively), and (e) the multiplicative coherent model, where each row (or column) is obtained by multiplying another row (or column, respectively) by a constant value. In this paper, we continue to exploit the most general type-(f) linear coherent model [7] (see Figure 1), in which each row is obtained by multiplying another row by a constant value and then adding a constant. We further assume that bi-clusters are arbitrarily positioned and may overlap each other [15]. The most biologically meaningful types of bi-clusters to be sought should map to the ultimate purpose of identifying groups of genes that co-participate in certain genetic regulatory process. For example, housekeeping genes are those that constitutively express in most conditions, and they could be identified in the first two bi-cluster models (a) and (b). Most of the existing bi-clustering algorithms seek either type-(d) additive bi-clusters or type-(e) multiplicative bi-clusters [7]. Mathematically, the type-(f) linear coherent model is strictly more general than all the other five models.

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)

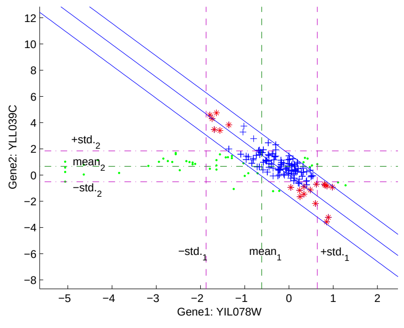
x	y	z	w
2.0	4.0	3.0	5.0
1.5	2.5	2.0	3.0
2.3	4.3	3.3	5.3
4.5	8.5	6.5	10.5

(f)

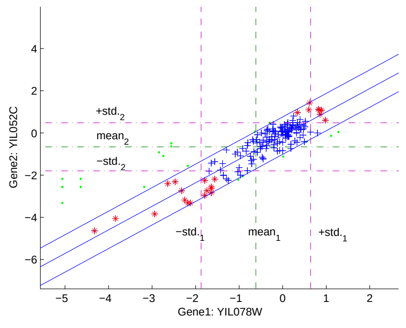
Fig. 1. The six different bi-cluster types.

The key idea in our new algorithm, LinCoh, for finding type-(f) linear coherent bi-clusters is illustrated in Figure 2. Essentially, a pair of genes participates in a linear coherent bi-cluster must be evidenced by a non-trivial subset of samples in which these two genes are co-up-regulated (or co-down-regulated). Therefore, the scatter plot of their pairwise expression levels, see Figure 2, where every point (x, y) represents a sample in which the two genes have expression levels x and y respectively, must show a diagonal band with a sufficient number of sample points. The LinCoh algorithm starts with composing this non-trivial supporting sample set for each gene pair, then to cluster these so-called *outer* sample sets. Each outer sample set cluster, together with the associated genes and *inner* samples, is filtered to produce a final bi-cluster.

We compare our LinCoh algorithm to four most popular bi-clustering algorithms: Cheng and Church's algorithm named after CC [3]; the order preserving sub-matrix algorithm denoted as OPSM [2]; the iterative signature algorithm denoted as ISA [10]; and the maximum similarity bi-clustering algorithm denoted as MSBE [14]. The first three algorithms have been selected and implemented



(a) Negative correlation.



(b) Positive correlation.

Fig. 2. (a) illustrates two yeast genes *YIL078W* and *YLL039C* that have negative expression correlation under a subset of conditions; the red conditions provide a stronger evidence than the blue conditions, whereas the green conditions do not suggest any correlation. Similarly in (b), genes *YIL078W* and *YIL052C* show a positive expression correlation.

in a survey [17]. Cheng and Church defined a merit score called *mean squared residue* to evaluate the quality of a bi-cluster, and CC is a greedy algorithm for finding bi-clusters of score no less than a given threshold [3]. OPSM is a heuristic algorithm attempting to find within a gene expression matrix the sub-matrices, *i.e.* bi-clusters, in each of which the genes have the same linear ordering of expression levels [2]. Another bi-cluster quality evaluation scheme is proposed in [10] using gene and condition signatures, and the ISA is proposed for finding the corresponding good quality bi-clusters. In particular, a randomized ISA is put in place when the prior information of the expression matrix is not available. The last algorithm, MSBE, is the first polynomial time bi-clustering algorithm that finds optimal solutions under certain constraints [14].

The rest of the paper is organized as follows: Section 2 presents the details of our LinCoh algorithm. In Section 3, we first introduce the quality measurements for bi-clustering results; then describe how synthetic datasets were generated, followed by the bi-clustering results and discussion; lastly we present the two real datasets on yeast and e.coli respectively, as well as the bi-clustering results and discussion. We conclude the paper in Section 4 with some remarks on the advantages and disadvantages of our LinCoh algorithm.

2 The LinCoh algorithm

Let $E(G, S)$ be an $n \times m$ gene expression data matrix, where $G = \{1, 2, \ldots, n\}$ is the set of gene indices and $S = \{1, 2, \ldots, m\}$ is the set of sample (condition) indices. Its element e_{ij} is the expression level of gene i in sample j. Our LinCoh algorithm consists of two major steps, described in the next two subsections.

2.1 Step one: establishing pairwise gene relations

For each pair of genes $p, q \in \{1, 2, \ldots, n\}$, we plot their expression levels in all samples in a 2D plane, as shown in Figure 2, where a point (x, y) represents a sample in which gene p has expression level x and gene q has expression level y. The goal of this step is to detect a correlation between every pair of genes on a subset of samples, if any. Such a subset of samples must evidence the correlation, in the way that the two genes are co-up-regulated (or co-down-regulated) in these samples [13]. We define a *beam* $B_{\theta,\beta,\gamma}$ in the 2D plane to be the set of points on the plane that are within distance $\frac{1}{2}\beta$ to a straight line that depends on θ and γ. Here θ is the *beam angle*, β is the *beam width*, and γ is the *beam offset*. They are all search parameters, but we are able to pre-determine some best values or ranges of values for them.

Let μ_p and σ_p (μ_q and σ_q) denote the mean expression level of gene p (q, respectively) across all samples and the standard deviation. To identify the subset of supporting samples for this gene pair, $S_{\theta,\beta,\gamma} = S \cap B_{\theta,\beta,\gamma}$, we seek for a beam $B_{\theta,\beta,\gamma}$ in the 2D plane that aligns approximately the main diagonal (or the antidiagonal) of the rectangle defined by $\{(\mu_p - \sigma_p, \mu_q - \sigma_q), (\mu_p + \sigma_p, \mu_q - \sigma_q), (\mu_p + \sigma_p, \mu_q + \sigma_q), (\mu_p - \sigma_p, \mu_q + \sigma_q)\}$. Such an approximate alignment optimizes the following objective function, which robustly leads to good quality bi-clusters:

$$\begin{aligned} &\max_{\theta,\beta,\gamma} \; W_{S_{\theta,\beta,\gamma}} \cdot D_{S_{\theta,\beta,\gamma}} \\ \text{subject to: } &\left| corr\big(E(p, S_{\theta,\beta,\gamma}), E(q, S_{\theta,\beta,\gamma})\big)\right| \geq t_{cc}. \end{aligned}$$

In the above maximization problem, $D_{S_{\theta,\beta,\gamma}}$ is the vector of the Euclidean distances of the samples inside the beam, *i.e.* $S_{\theta,\beta,\gamma}$, to the line passing through (μ_p, μ_q) and perpendicular to (called the *midsplit line* of) the beam center line; $W_{S_{\theta,\beta,\gamma}}$ is a weight vector over the samples in $S_{\theta,\beta,\gamma}$, and we use Shepard's function $w_j = d_j^r / \sum d_j^r$ with parameter $r \geq 0$ to weight sample $j \in S_{\theta,\beta,\gamma}$ (to weight more on distant samples but less on samples nearby the midsplit line); In the constraint, $|corr(\cdot, \cdot)|$ is the absolute correlation coefficient between the two genes p and q, calculated over only the samples in $S_{\theta,\beta,\gamma}$; t_{cc} is a pre-determined correlation threshold.

The output of the maximization problem is $S_{\theta,\beta,\gamma}$, which is either empty, indicating that no meaningful relationship between the two genes was found, or otherwise a subset of samples that evidence a meaningful correlation between genes p and q. According to our extensive preliminary experiments, the bi-clustering results are of high quality when the correlation threshold t_{cc} is larger than 0.75; and it is set to 0.90 and 0.75, respectively, on synthetic datasets and real datasets.

We implement a heuristic process to search for the beam, whose center line is initialized to be the line passing through the main diagonal (for positive correlation) or the antidiagonal (for negative correlation) of the rectangle in the expression plot. The beam width β is fixed at a certain portion of $4\sigma_p\sigma_q/\sqrt{{\sigma_p}^2+{\sigma_q}^2}$; again supported by the preliminary experiments, a constant portion in between 0.8 and 1.0 is sufficient to capture most meaningful correlations; and we fix it at 0.8 for both synthetic datasets and real datasets. That is, $\beta = 0.8 \times 4\sigma_p\sigma_q/\sqrt{{\sigma_p}^2+{\sigma_q}^2}$. To determine the beam angle θ in the positive correlation case, we define the search axis to be the midsplit line of the main diagonal; a small interval is placed on the search axis centering at (μ_p, μ_q), which is for a pivot point to float within; around each position of the pivot point, whose distance to the center point (μ_p, μ_q) is denoted as γ, different angles (the θ) are searched over to find a beam center line; each resultant beam is tested for the constraint satisfaction in the maximization problem, and discarded otherwise; among all those beams that satisfy the constraint, the one maximizing the objective function is returned as the target beam.

For evaluating the objective function, we have tested multiple values of r and found that 0 gives the most robust bi-clustering results. Therefore, r is set to 0 as default. For each sample j inside the beam, its distance d_j to the mid-split line of the beam center line is rounded to 0 or 1 using a threshold of $\sqrt{{\sigma_p}^2+{\sigma_q}^2}$. When the target beam is identified, though might not be the true optimum to the objective function, the sample set $S_{\theta,\beta,\gamma}$ is further partitioned into *outer* sample set (containing the samples with distance d_j rounded to 1) and *inner* sample set (containing the rest). Gene pairs, together with non-empty outer and inner sample sets, are sent to Step two for clustering.

2.2 Step two: sample set clustering

Step one generates an outer sample set and an inner sample set for each gene pair. In this step, two $n \times n$ matrices are constructed: in the *outer matrix* M^o, the element m^o_{pq} is the outer sample set for gene pair p and q; likewise, in the *inner matrix* M^i, the element m^i_{pq} is the inner sample set for gene pair p and q. We next process these two matrices to robustly find bi-clusters.

First we want to filter out small outer sample sets that indicate insignificant correlations for gene pairs. To this purpose, we select roughly the largest 0.15% outer sample sets among all for clustering, which are of 99.7% confidence. This confidence level is set after testing on a randomly generated datasets, with 68%, 95%, 99.7% confidence levels according to the 68-95-99.7 rule. Observing that two disjoint gene pairs could have the same outer sample sets but very different expression patterns, simply using outer sample sets to construct bi-clusters might lead to meaningless bi-clusters. In our LinCoh algorithm, genes are used as bridges to group similar outer sample sets to form bi-clusters, since linear correlation is transitive. We first define the similarity between two outer sample sets m^o_{pq} and m^o_{rs} as $\mathrm{sim}(m^o_{pq}, m^o_{rs}) = |m^o_{pq} \cap m^o_{rs}|/|m^o_{pq} \cup m^o_{rs}|$, which is a most popular measure in the literature. Next, we rank genes in the descending order of the number of associated non-empty outer sample sets. Iteratively, the gene at the head of this list is used as the *seed* gene, to collect all its associated (non-empty) outer sample sets. These outer sample sets are partitioned into clusters using a density based clustering algorithm similar to DBSCAN [5], and the densest cluster is returned, which is defined as a cluster whose central point has the most close neighbors (see the pseudocode in the Appendix). An initial bi-cluster is formed on the union S_1 of the outer sample sets in the densest cluster, and the set G_1 of the involved genes.

The quality of the bi-cluster (G_1, S_1) is evaluated by its *average absolute correlation coefficient*,

$$\text{aacc}(G_1, S_1) = \frac{\sum_{p,q \in G_1} |corr(E(p, S_1), E(q, S_1))|}{(|G_1|^2 - |G_1|)}. \tag{1}$$

The initial bi-cluster (G_1, S_1) is refined in three steps to locally improve its quality. In the first step, all samples in S_1 are sorted in decreasing frequency of occurrence in all the outer sample sets of the seed gene; the lowest ranked sample is removed if this removal improves the quality of the bi-cluster, or otherwise the first step is done. Secondly, every gene in G_1 is checked to see whether its removal improves the quality of the bi-cluster, and if so it is removed from G_1. At the end, if the minimum gene pairwise absolute correlation coefficient of the bi-cluster is smaller than a threshold, the bi-cluster is considered as of low quality and discarded. By examining values from 0.50 to 0.99, our preliminary experiments showed that a high threshold in between 0.7 and 0.9 is able to ensure good quality bi-clusters, and it is set to 0.8 in all our final experiments. In the last step of bi-cluster (G_1, S_1) refinement, the inner sample sets of the gene pairs from G_1 are collected; their samples are sorted in decreasing frequencies in these inner sample sets; using this order, samples are added to S_1 as long as their addition passes the 0.8 minimum pairwise absolute correlation coefficient. A final bi-cluster (G_1, S_1) is thus produced.

Subsequently, all genes from G_1 are removed from the gene list, and the next gene is used as the seed gene for finding the next bi-cluster. The process iterates till the gene list becomes empty. We remark that a gene can participate in multiple bi-clusters, but it serves as a seed gene at most once. At the end, when two bi-clusters overlap more than 60% area, the one of smaller size is treated as redundant and discarded [13]. A pseudocode of our LinCoh algorithm 1 is provided in the Appendix for the interested readers.

3 Results and discussion

We examine the LinCoh algorithm, and make comparisons with four other existing bi-clustering algorithms, CC, OPSM, ISA, and MSBE (their parameter settings follow the previous works [17,14]), on many synthetic datasets and two real gene expression microarray datasets on *Saccharomyces cerevisiae* (yeast) and *Escherichia coli* (e.coli) respectively. Essentially, synthetic datasets are used for evaluating absolute performance, since we know the ground truth; while the real datasets are mainly used for evaluating relative performance. Consequently we have different sets of performance measurements on synthetic and real datasets.

On synthetic datasets, bi-clustering algorithms are evaluated on their ability to recover the implanted (true) bi-clusters. Prelić's gene match score and overall match score [17] are adopted. Let $\mathcal{C}$ and $\mathcal{C}^*$ denote the set of output bi-clusters from an algorithm and the set of true bi-clusters for a dataset. The gene match score of $\mathcal{C}$ with respect to the target $\mathcal{C}^*$ is defined as $\text{score}_G(\mathcal{C}, \mathcal{C}^*) = \frac{1}{|\mathcal{C}|} \sum_{(G_1,S_1)\in\mathcal{C}} \max_{(G_1^*,S_1^*)\in\mathcal{C}^*} \frac{|G_1 \cap G_1^*|}{|G_1 \cup G_1^*|}$, which is essentially the average of the maximum gene match scores of bi-clusters in $\mathcal{C}$ with respect to the target bi-clusters. Similarly, the sample match score $\text{score}_S(\mathcal{C}, \mathcal{C}^*)$ can be defined by replacing gene sets with the corresponding sample sets in the above. The overall match score is then defined as their geometric mean, *i.e.*

$$\text{score}(\mathcal{C}, \mathcal{C}^*) = \sqrt{\text{score}_G(\mathcal{C}, \mathcal{C}^*) \times \text{score}_S(\mathcal{C}, \mathcal{C}^*)}.$$

On real datasets, the bi-clusters discovered by an algorithm are mapped to known biological pathways, defined in the GO functional classification scheme [1], the KEGG pathways [11], the MIPS yeast functional categories [18] (for yeast dataset), and the EcoCyc database [12] (for e.coli dataset), to obtain their *gene functional enrichment score* as implemented in [13]. The average absolute correlation coefficients (aacc's) of the discovered bi-clusters are also used to compare different algorithms.

3.1 Synthetic datasets

Noise resistance test: This experiment examines how well a bi-clustering algorithm can recover implanted bi-clusters. We follow Prelić's testing strategy to first generate a 100×50 background matrix (*i.e.*, 100 genes and 50 samples), using a standard normal distribution for the matrix elements; we then embed ten 10×5 non-overlapping linear coherent bi-clusters along the diagonal; later for each vector of the five expression values, the first two of them are set to down-regulated, the last two are set to up-regulated, and the middle one is non-regulated; lastly, we add noise to the embedded bi-clusters at six different noise levels ($\ell = 0.00, 0.05, 0.10, 0.15, 0.20, 0.25$) by perturbing the entry values so that the resultant values are ℓ away from the original values. The generation is repeated ten times to give ten matrices.

The same simulation process is done to generate synthetic datasets containing additive bi-clusters, when we compare the bi-clustering algorithms on their performance to recover additive bi-clusters only (which is a special case of linear coherent bi-clusters).

Figure 3 shows the gene match scores of all five bi-clustering algorithms at six different noise levels, on their performance of recovering linear coherent bi-clusters and additive bi-clusters, respectively. Their overall match scores and gene discovery rates (defined as the percentage of genes in the output bi-clusters over all the genes in the true bi-clusters) can be found in Figures 8 and 9 in the Appendix. In terms of match scores, Figures 3 and 8 clearly show that our LinCoh outperformed all the other four algorithms, ISA ranked the second, and the other three performed quite poorly. In terms of gene discovery rate, again LinCoh outperformed all the other four algorithms. We remark that gene discovery rate can be trivially lifted up by simply output more bi-clusters. It is not a main measure used in this work, but a useful measure in conjunction with match scores.

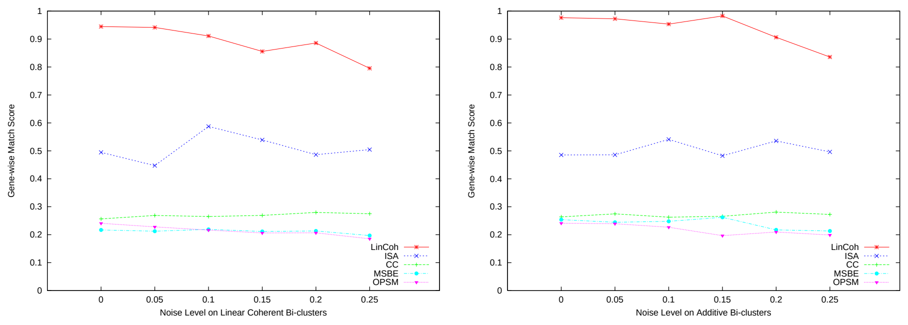


Fig. 3. The gene match scores of the five algorithms on recovering linear coherent bi-clusters and additive bi-clusters at six different noise levels.

Overlapping test: Individual genes can participate in multiple biological processes, yielding bi-clusters that overlap with common genes in an expression matrix. Bi-clusters might also overlap with a subset of samples. This experiment is designed to examine the ability of different bi-clustering algorithms to recover overlapping bi-clusters. As before, we consider type-(f) linear coherent bi-clusters and type-(d) additive bi-clusters, at a fixed noise level of $\ell = 0.1$.

Again, ten 100×50 background matrices are generated using a standard normal distribution for the matrix elements; into each of them, two 10×10 bi-clusters are embedded, overlapping with each other by one of the following six cases: 0×0, 1×1, 2×2, 3×3, 4×4, and 5×5. Previous simulation studies suggested to replace the matrix elements in the overlapped area with a random value; we expect, however, these overlapping genes to obey a reasonable logic such as the AND gate and the OR gate leading to a *union* and an *additive* behavior. Therefore, in the union overlap model, the matrix entries in the overlapped area preserve linear coherency in both bi-clusters (consequently, the overlapped area extends its linear coherency into both bi-clusters on those samples in the overlapped area); and in the additive overlap model, these entries take the sum of the gene expression levels from both bi-clusters.

Figure 4 shows the gene match scores of the five bi-clustering algorithms in this experiment. Their overall match scores and gene discovery rates under the union overlap model are plotted in Figures 10 and 11 in the Appendix.

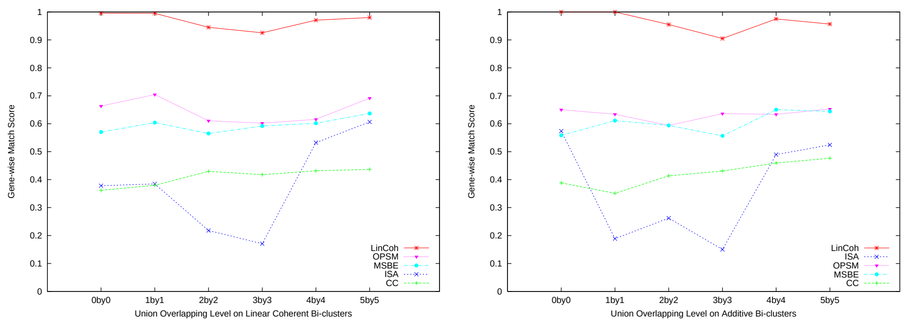


Fig. 4. The gene match scores of the five algorithms for recovering the overlapping linear coherent and additive bi-clusters, under the union overlap model.

The results of the additive overlap model are in Figures 12, 13, and 14 in the Appendix. From all these results, one can see that our LinCoh outperformed the other four algorithms; OPSM and MSBE performed worse, but similarly to each other; CC performed the worst; and ISA demonstrated varying performance.

3.2 Real datasets

The yeast dataset is obtained from [8], containing 2993 genes on 173 samples; the e.coli dataset (version 4 built 3) is from [6], which contains initially 4217 genes on 264 samples. Genes with small expression deviations were removed from the second dataset, giving rise to 3016 genes. Such a process ensures that all five bi-clustering algorithms can run on the dataset. In particular, it took two weeks for LinCoh to run on each dataset using a 2.2GHz CPU node of 2.5GB memory. The performance of an algorithm on these two real datasets is measured in gene functional enrichment score [13]. First, the P-value of each output bi-cluster is defined using its most enriched functional class (biological process).

The probability of having r genes of the same functional class in a bi-cluster of size n from a genome with a total of N genes can be computed using the hypergeometric function, where p is the percentage of that functional class of genes over all functional classes of genes encoded in the whole genome. Numerically [13],

$$Pr(r|N,p,n) = \binom{pN}{r} \cdot \binom{(1-p)N}{n-r} / \binom{N}{n}.$$

Such a probability is taken as the P-value of the output bi-cluster enriched with genes from that functional class [13]. The smallest P-value over all functional classes is defined as the P-value of the output bi-cluster — the smaller the P-value of a bi-cluster the more likely its genes come from the same biological process. For each algorithm, we calculate the fraction of its output bi-clusters whose P-values are smaller than a significance cutoff α.

Figure 5 compares the five algorithms using six different P-value cutoffs, evaluated on the GO database. Results on the KEGG, MIPS, and Regulon databases

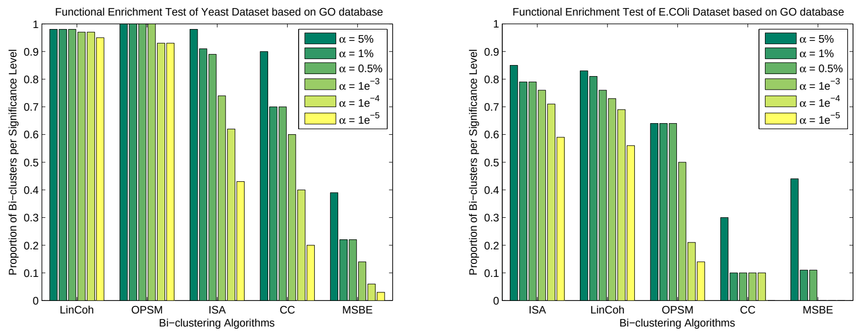


Fig. 5. Portions of discovered bi-clusters by the five algorithms on the two real datasets that are significantly enriched the GO biological process, using six different P-value cutoffs.

are in Figures 15 and 16 in the Appendix. All these results show that our LinCoh consistently performed well; OPSM and ISA did not perform consistently on the two datasets across databases; and that MSBE and CC did not perform as well as the other three algorithms.

One potential issue with the P-value based performance measurement is that P-values are sensitive to the bi-cluster size [13]; in general, larger bi-clusters tend to produce more significant P-values. Table 1 in the Appendix summarizes the statistics of the bi-clusters produced by the five algorithms. The last column in the table records the numbers of unique functional terms enriched by the produced bi-clusters. On yeast dataset, when measured by the gene enrichment significance test, OPSM performed very well (Figure 5, left); yet its bi-clusters only cover one functional term on the GO and KEGG databases and two terms on MIPS database. Such a phenomenon suggests that its bi-clustering result is biased to a group of correlated genes, missed by the P-value based significance test. Furthermore, we generated all the gene pairs with absolute correlation coefficient greater than or equal to 0.8 over all the samples for both the yeast and e.coli datasets. Table 2 in the Appendix shows the numbers of common GO terms and their counts. Among these strongly correlated gene pairs, many do not even have one common GO term. Table 3 in the Appendix shows the top 10 counted common GO terms (full table can be found at '`http://www.cs.ualberta.ca/~ys3/LinCoh`').

The above two potential issues hint that the P-value based evaluation is meaningful but has limitations. We propose to use the average absolute correlation coefficient over all gene pairs in a bi-cluster defined in Eq. (1) as an alternative assessment of the quality of a linear coherent bi-cluster. Figure 6 shows the box plot of these correlation values for the bi-clusters produced by the five algorithms on the two real datasets. From the figure, one can see that our LinCoh and OPSM significantly outperformed the other three algorithms.

Additionally, the minimum absolute correlation coefficient over all gene pairs in a bi-cluster can also be adopted as a quality measurement. Figure 17 in the Appendix shows these results.

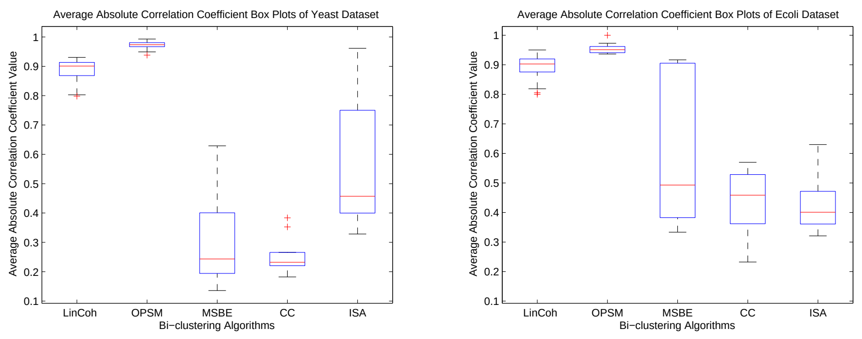


Fig. 6. Box plots of the average absolute correlation coefficients obtained by the five bi-clustering algorithms on yeast and e.coli datasets, respectively.

Figure 6 shows that OPSM produced bi-clusters with very high linear coherence. But the numbers of samples in its bi-clusters are much smaller than those in LinCoh's bi-clusters, as shown in Table 1 in the Appendix (tens versus hundreds). This suggests that very closely interacting gene pairs can have small empirical correlation coefficients on a subset of samples, largely due to noise and measurement errors. In fact, there is always a trade-off between bi-cluster coherence and its size. Thus, to compare algorithms in a less sample-size biased way, we replaced for each bi-cluster its average absolute correlation coefficient by the 99% confidence threshold using the number of samples in the bi-cluster [19], and box plotted these values in Figure 18 in the Appendix. They show much more comparable performance between LinCoh and OPSM.

4 Conclusions

In this paper, we proposed a new bi-clustering algorithm, LinCoh, for finding linear coherent bi-clusters in a gene expression matrix. The algorithm has two important steps, beam detection for pairwise gene correlations and density based outer sample set clustering. Our experiments on synthetic and real datasets demonstrate that LinCoh consistently performed well. Using real datasets, we also showed some limitations of the widely adopted functional enrichment measurement, and proposed to use average absolute correlation coefficient as an alternative measure for bi-clustering quality. With its outperformance over the compared four popular algorithms, LinCoh can serve as another useful tool for microarray data analysis, including bi-clustering and genetic regulatory network inference.

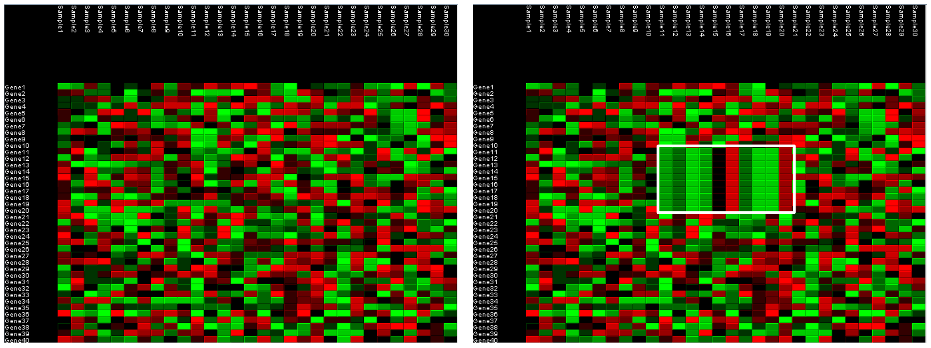
One disadvantage of LinCoh is its large memory and extensive computing time requirement, due to constructing the outer and inner sample set matrices. It takes $O(n^2 m_p)$ to compute the sample set matrices where n is the number of genes and m_p is the number of parameters θ, β and γ. The memory required for storing the matrices is $O(n^2 m_s)$ where n is the number of genes and m_s is the average size of the sample set elements. It takes weeks and up to 1 Gigabyte memory to run experiments on the e.coli dataset. Improvements in beam detection and sample set clustering can also achieve significant speed-ups.

References

1. M. Ashburner, C. A. Ball, and J. A. Blake *et. al.* Gene ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.
2. A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini. Discovering local structure in gene expression data: The order-preserving sub-matrix problem. In *RECOMB'02*, pages 49–57, 2002.
3. Y. Cheng and G. M. Church. Biclustering of expression data. In *ISMB'00*, pages 93–103, 2000.
4. M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *PNAS*, 95:14863–14868, 1998.
5. M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD'96*, pages 226–231, 1996.
6. J. J. Faith, M. E. Driscoll, and V. A. Fusaro *et al.* Many microbe microarrays database: uniformly normalized Affymetrix compendia with structured experimental metadata. *Nucleic Acids Research*, 36:D866–D870, 2008.
7. 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.
8. A. P. Gasch, P. T. Spellman, and C. M. Kao *et al.* Genomic expression programs in the response of yeast cells to environmental changes. *Nucleic Acids Research*, 11:4241–4257, 2000.
9. J. A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67:123–129, 1972.
10. J. Ihmels, S. Bergmann, and N. Barkai. Defining transcription modules using large scale gene expression data. *Bioinformatics*, 20:1993–2003, 2004.
11. M. Kanehisa. The KEGG database. *Novartis Foundation Symposium*, 247:91–101, 2002.
12. I. M. Keseler, J. Collado-Vides, and S. Gama-Castro *et al.* EcoCyc: a comprehensive database resource for *escherichia coli*. *Nucleic Acids Research*, 33:D334–D337, 2005.
13. G. Li, Q. Ma, H. Tang, A. H. Paterson, and Y. Xu. QUBIC: A qualitative biclustering algorithm for analyses of gene expression data. *Nucleic Acids Research*, 37:e101, 2009.
14. X. Liu and L. Wang. Computing the maximum similarity bi-clusters of gene expression data. *Bioinformatics*, 23:50–56, 2006.

15. S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *Journal of Computational Biology and Bioinformatics*, 1:24–45, 2004.
16. B. Mirkin. Mathematical classification and clustering. *Kluwer Academic Publishers*, 1996.
17. A. Prelić, S. Bleuler, P. Zimmermann, and A. Wille. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, 22:1122–1129, 2006.
18. A. Ruepp, A. Zollner, and D. Maier *et al.* The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Research*, 32:5539–5545, 2004.
19. D. Shen and Z. Lu. Computation of correlation coefficient and its confidence interval in SAS. http://www2.sas.com/proceedings/sugi31/170-31.pdf.
20. P. Tamayo, D. Slonim, and J. Mesirov *et al.* Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *PNAS*, 96:2907–2912, 1999.
21. 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.

Appendix



(a) Unobvious Bi-cluster. (b) Obvious Bi-cluster.

Fig. 7. An example of a constant row bi-cluster in the gene expression matrix. (a) shows a gene expression matrix without any obvious bi-clusters; (b) shows after swapping rows and columns, a constant row bi-cluster becomes salient.

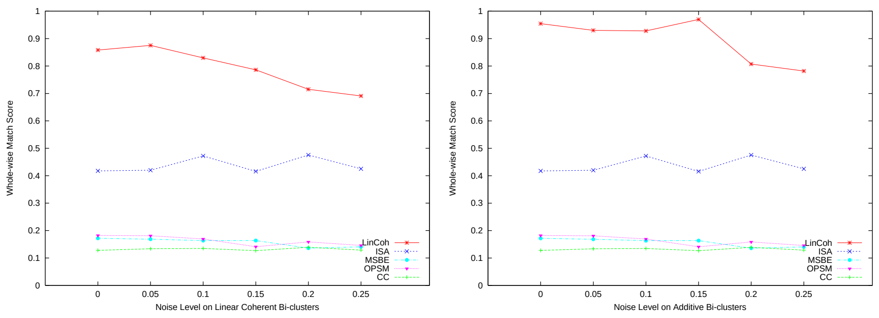


Fig. 8. The overall match scores of the five algorithms for recovering linear coherent and additive bi-clusters, at six different noise levels.

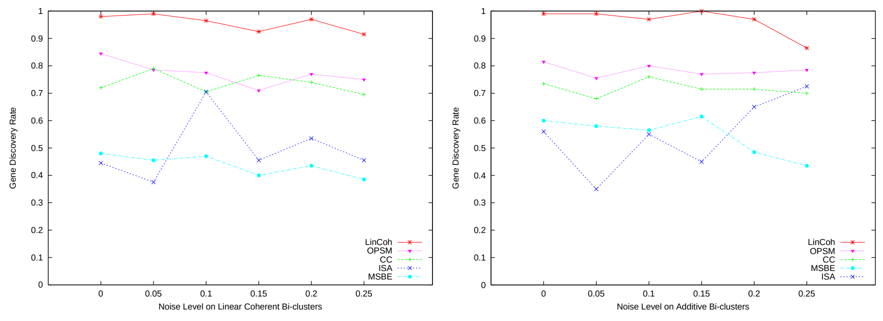


Fig. 9. The gene discovery rates of the five algorithms for recovering linear coherent and additive bi-clusters, at six different noise levels.

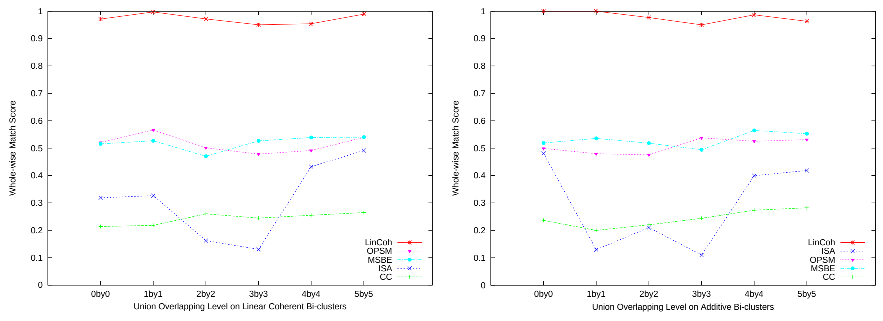


Fig. 10. The overall match scores of the five algorithms for recovering linear coherent and additive bi-clusters, under six different amounts of overlap using the union overlap model.

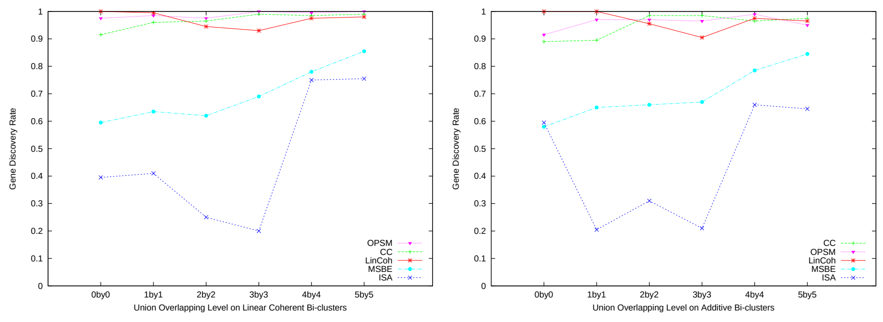


Fig. 11. The gene discovery rates of the five algorithms for recovering linear coherent and additive bi-clusters, under six different amounts of overlap using the union overlap model.

Algorithm 1 The LinCoh Algorithm

```
Input An $n \times m$ real value matrix $A(I, J)$, $T_{close}$, $T_{minCC}$
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
    $M_O(i, j) = NULL, \quad M_I(i, j) = NULL;$
    $\theta_{rec} = NULL, \quad \beta_{rec} = NULL, \quad \gamma_{rec} = NULL;$
    for A set of beam parameters ($\theta$, $\beta$, $\gamma$) do
      if $W_{S_{outer(\theta,\beta,\gamma)}} \cdot D^T_{S_{outer(\theta,\beta,\gamma)}} > |M_O(i, j)|$ then
        $M_O(i, j) = S_{outer(\theta,\beta,\gamma)};$
        $\theta_{rec} = \theta, \quad \beta_{rec} = \beta, \quad \gamma_{rec} = \gamma;$
      end if
    end for
    $M_I(i, j) = S_{inner(\theta_{rec},\beta_{rec},\gamma_{rec})};$
  end for
end for

for $i = 1$ to n do
  for $j = i + 1$ to n do
    if $M_O(i, j) < \mu_{ss} + \alpha \cdot \sigma_{ss}$ then
      $M_O(i, j) = NULL$, $M_I(i, j) = NULL;$
    end if
  end for
end for

for $i = 1$ to n do
  $SS_i = \bigcup_{j \in J}(M_O(i, j));$
end for
$GeneList_{ss}$ = DescendSort($Genes$) based on $|SS_i| \neq NULL$ of each $i \in I$;
$BiclusterPool = NULL;$
while $GeneList_{ss} \neq EMPTY$ do
  $SeedGene = Pop(GeneList_{ss});$
  Construct similarity matrix $Matrix_{ss}$ for $SS_{seedGene}$ elements based on
  $M_S(SS_i, SS_j) = \frac{|SS_i \cap SS_j|}{|SS_i \cup SS_j|};$
  Find the centroid sample set $SS_{centroid}$ that has the most *close* ($M_S(S_i, S_j) \geq$
  $T_{close}$) neighbors, $SS_{neighbors}$;
  $GenePool = \bigcup(G_i \in G_{SS_{centroid}} \bigcup G_{SS_{neighbors}});$
  $SamplePool = \bigcup(S_j \in SS_{centroid} \bigcup SS_{neighbors});$
  $BiCluster_{initial} = A(GenePool, SamplePool);$
  $BiCluster_{refined} = RefineBicluster(BiCluster_{initial})$
  if MinAbsCC($BiCluster_{refined}$) $\geq T_{minCC}$ then
    BiclusterPool.add($BiCluster_{refined}$);
  end if
end while
$Biclusters_{final} = RedundantRemoval(BiclusterPool)$
```

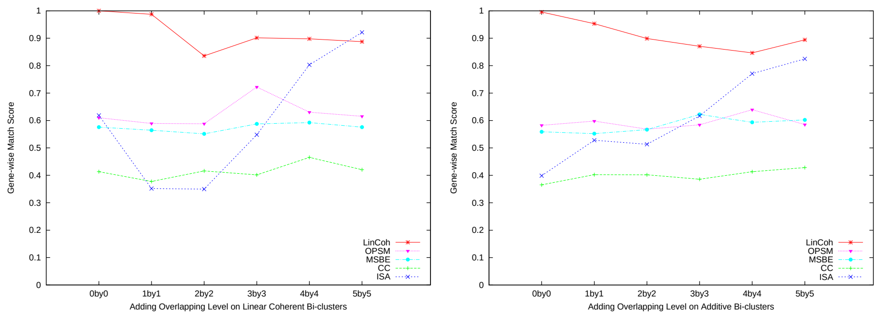


Fig. 12. The gene match scores of the five algorithms for recovering linear coherent and additive bi-clusters, under six different amounts of overlap using the additive overlap model.

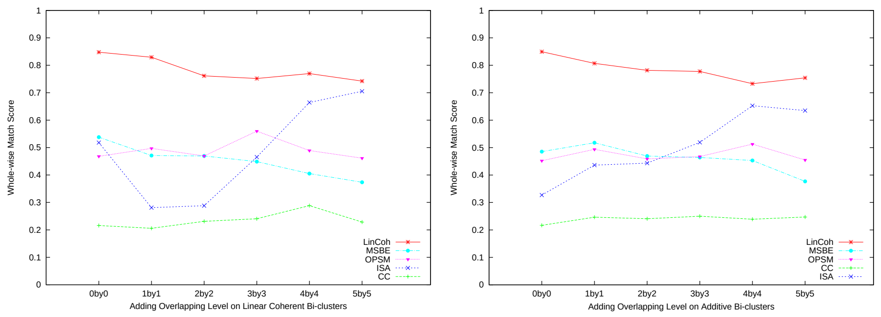


Fig. 13. The overall match scores of the five algorithms for recovering linear coherent and additive bi-clusters, under six different amounts of overlap using the additive overlap model.

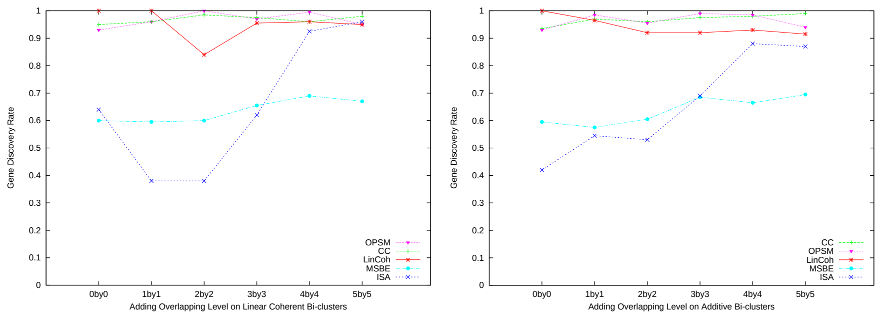


Fig. 14. The gene discovery rates of the five algorithms for recovering linear coherent and additive bi-clusters, under six different amounts of overlap using the additive overlap model.

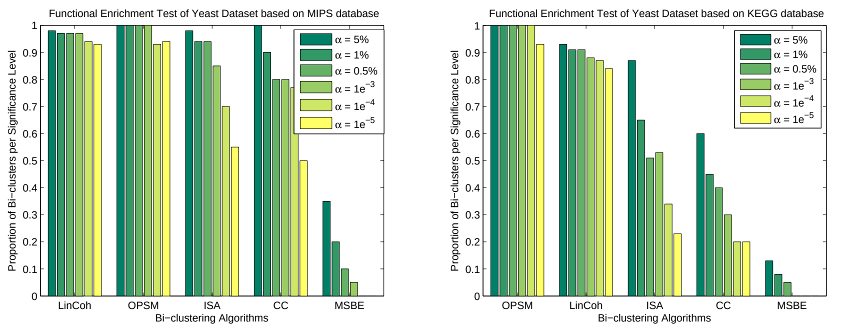


Fig. 15. Portions of yeast bi-clusters that are significantly enriched over different P-values in the MIPS pathway and KEGG pathway.

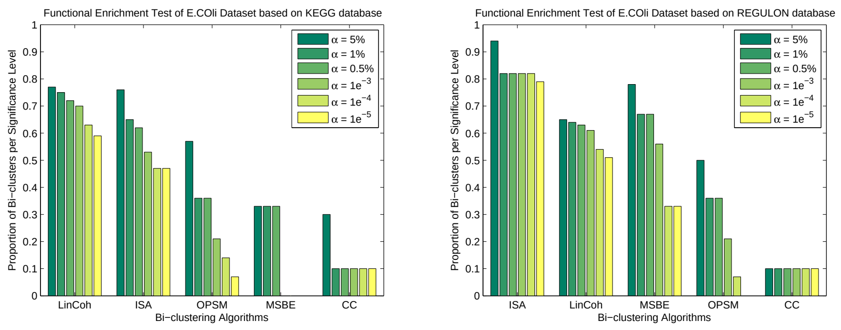


Fig. 16. Portions of e.coli bi-clusters that are significantly enriched over different P-values in the KEGG pathway and experimentally verified regulons.

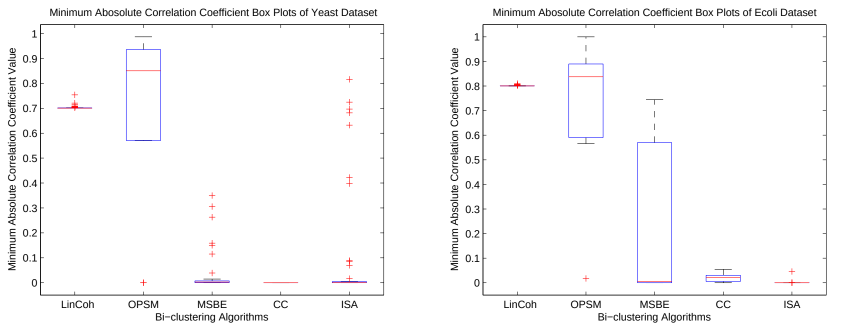


Fig. 17. The box plots of minimum absolute correlation coefficients of the bi-clusters produced by the five algorithms on the yeast and e.coli datasets.

	#Bi-clusters	$\mu_{\|gene\|}$	$\sigma_{\|gene\|}$	$\mu_{\|sample\|}$	$\sigma_{\|sample\|}$	#Unique terms enriched (GO, KEGG, MIPS/regulons)
yeast:						
LinCoh	100	61.84	38.43	133.09	18.09	5, 7, 5
ISA	47	67	34.54	8.4	1.78	15, 13, 18
OPSM	14	423.29	728.95	9.07	5.14	1, 1, 2
MSBE	40	19.25	8.32	18.68	8.22	8, 4, 6
CC	10	297.7	304.18	60.8	23.46	6, 4, 8
e.coli:						
LinCoh	100	9.63	7.66	141.43	34.04	24, 24, 22
ISA	34	124.21	42.18	13.88	6.11	11, 10, 13
OPSM	14	419.29	744.35	8.93	4.8	8, 4, 5
MSBE	9	82.67	18.1	80.22	19.18	1, 3, 4
CC	10	309.9	950.15	31.4	81.74	2, 2, 2

Table 1. Statistics of different algorithms' bi-clustering results and the numbers of functional terms enriched on different databases.

Term count	yeast	e.coli
0	909	2680
1	18860	3485
2	7898	1533
3	1839	490
4	165	239
5	30	52
6	6	18
7	4	0
Overall	28802	5817

Table 2. For all the gene pairs with absolute correlation coefficient ≥ 0.8, the number of pairs that have between 0 and 7 common GO terms.

yeast		e.coli	
GO term	Count	GO term	Count
GO:0006412	8353	GO:0006412	811
GO:0000723	1920	GO:0008652	680
GO:0000027	1615	GO:0001539	388
GO:0000028	1070	GO:0006810	317
GO:0006365	969	GO:0006355	234
GO:0006413	893	GO:0006811	195
GO:0030488	782	GO:0006865	183
GO:0006364	720	GO:0006260	127
GO:0030490	683	GO:0046677	115
GO:0006360	424	GO:0008152	111

Table 3. The top 10 gene pairs' common GO terms and their counts.

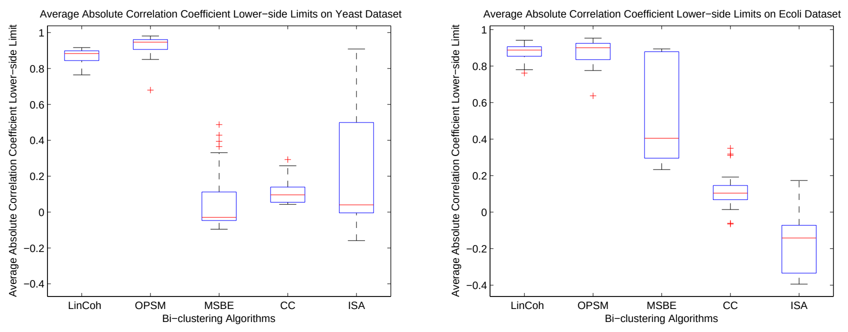


Fig. 18. The box plots of the 99% confidence thresholds of the average absolute correlation coefficients of the bi-clusters, using the number of samples in each bi-cluster, produced by the five algorithms on the yeast and e.coli datasets.