Sparse Learning based Linear Coherent Bi-clustering

Yi Shi[1⋆], Xiaoping Liao[2], Xinhua Zhang[1], Guohui Lin[1], and Dale Schuurmans[1]

[1]Department of Computing Science,
University of Alberta, Edmonton, Alberta, Canada
[2]Department of Agricultural, Food and Nutritional Science,
University of Alberta, Edmonton, Alberta, Canada
{ys3,xliao2,xinhua2,guohui,daes}@ualberta.ca

Abstract. Clustering algorithms are often limited by an assumption that each data point belongs to a single class, and furthermore that all features of a data point are relevant to class determination. Such assumptions are inappropriate in applications such as gene clustering, where, given expression profile data, genes may exhibit similar behaviors only under some, but not all conditions, and genes may participate in more than one functional process and hence belong to multiple groups. Identifying genes that have similar expression patterns in a common subset of conditions is a central problem in gene expression microarray analysis. To overcome the limitations of standard clustering methods for this purpose, Bi-clustering has often been proposed as an alternative approach, where one seeks groups of observations that exhibit similar patterns over a subset of the features. In this paper, we propose a new bi-clustering algorithm for identifying linear-coherent bi-clusters in gene expression data, strictly generalizing the type of bi-cluster structure considered by other methods. Our algorithm is based on recent sparse learning techniques that have gained significant attention in the machine learning research community. In this work, we propose a novel sparse learning based model, SLLB, for solving the *linear coherent* bi-clustering problem. Experiments on both synthetic data and real gene expression data demonstrate the model is significantly more effective than current bi-clustering algorithms for these problems. The parameter selection problem and the model's usefulness in other machine learning clustering applications are also discussed. The Appendix of this paper can be found on `http://www.cs.ualberta.ca/~ys3/SLLB`.

Keywords: Bi-clustering, Microarray, Sparse Learning, Gene Expression, Linear Coherent

1 Introduction

Gene expression microarrays measure the expression levels of thousands of genes across multiple conditions (conditions are also often referred to as *samples*).

⋆ Corresponding author.

Identifying groups of genes that have similar expression patterns in a common subset of conditions is a central problem in gene expression microarray data analysis. Unfortunately, traditional clustering methods, such as those deployed in [5, 24, 25], are ill-suited to this purpose for two reasons: that genes may 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.

To overcome the limitations of standard clustering methods, *bi-clustering* [10, 18] has been proposed to identify groups of data points that exhibit similar patterns on a subset of features. The first work to apply bi-clustering to gene expression analysis is [4], which has motivated many other bi-clustering based approaches. Although the general bi-clustering problem is NP-hard [4], many papers have proposed heuristic methods for finding bi-clusters of different types. In particular, as illustrated in Figure 4 in Appendix, there are six different types of bi-clusters that have been sought in previous work, 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), (e) the multiplicative coherent model, where each row (or column) is obtained by multiplying another row (or column, respectively) by a constant value, and (f) the linear coherent model [7], in which each row (or column) is obtained by multiplying another row (or column) by a constant value and then adding a constant [23]. Mathematically, the linear coherent model (f) is strictly more general than the other five models, considered either row-wise or column-wise. In this paper, we design an algorithm that discovers linear coherent bi-clusters that are arbitrarily positioned and possibly even overlapping [16]. Note that, although bi-clusters cannot be simultaneously row-wise and column-wise linear coherent, one is usually more interested in clustering one dimension than the other [9, 2]. For example, in the case of gene expression analysis, the main purpose is to identify groups of *genes* that co-participate in certain genetic regulatory process, hence grouping conditions (samples) is only a secondary consideration. Most bi-clustering algorithms implicitly address non time series microarray data and only few address time series microarray data [17]. For time series data, the time lag between mRNA transcription and transcription factor translation needs to be considered. In this paper, we address non time series data.

The motivation for considering linear coherent bi-clusters for gene expression analysis specifically is illustrated in Figure 5 in Appendix [23]. The participation of a pair of genes 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). Due to data noise, the linear coherence exhibit beams rather than lines in a gene pairwise 2D plot [23, 9].

We compare our Sparse Learning based Linear Coherent Bi-clustering (SLLB) algorithm to seven representative bi-clustering algorithms that have been predominant in the field. The first method is a recent bi-clustering algorithm called QUBIC, it finds bi-clusters through employing a combination of qualitative (or

semi-quantitative) measures of gene expression data and a combinatorial optimization technique [14]. The second method is "Linear Coherent Bi-cluster Discovery via Beam Detection and Sample Set Clustering" (LinCoh) [23], which detects linear bi-clusters by first evaluating the correlation of gene pairs, and then clustering the sample sets that evidence the correlation. The third method is "Linear Coherent Bi-cluster Discovery via Line Detection and Sample Majority Voting" (LCBD) [22], which is the line detection version of LinCoh. Then, we compare to the maximum similarity bi-clustering algorithm (MSBE) [15], which is the first polynomial time bi-clustering algorithm that finds optimal solutions under certain constraints. Then, we compare to the iterative signature algorithm (ISA) [11], which is based on a bi-cluster quality evaluation scheme that uses gene and condition signatures. One advantage of this method is that it can handle incomplete data by imputing a randomized ISA in locations where the expression value is not available. Then, we compare to the order preserving sub-matrix algorithm (OPSM) [3], which attempts to find bi-clusters within a gene expression matrix that contains genes having the same linear ordering of expression levels. Finally, we compare to the method of Cheng and Church (CC) [4], which evaluates the quality of a bi-cluster by a proposed merit score called *mean squared residue*, and then applies a greedy algorithm to find bi-clusters with a score greater than some given threshold. the last three methods have all been highlighted and implemented in a recent survey [19].

The remainder of this paper is organized as follows. Section 2 first introduces the details of our SLLB method we propose. Then, Section 3 introduces the quality measurements we will use to assess the bi-clustering results, provides an experimental evaluation on synthetic data sets, and finally presents bi-clustering results on two real datasets, namely yeast and e.coli. Section 4 then concludes this work with some remarks on the advantages and disadvantages of the proposed SLLB algorithm.

2 Methods

The goal of this work is that given a matrix M (n observations $\times$ p features), find row-wise linear coherent bi-clusters so that each cluster exhibits row-wise linear coherence under a subset of common features.

Let us first consider a pair of $1 \times p$ observation vectors $\mathbf{m_{i:}}$ and $\mathbf{m_{j:}}$. Here $\mathbf{m_{i:}}$ is defined as the ith row vector of matrix M. Other row/column vectors appearing later in this section will be written in the same way. For a given subset of features we can always find the linear regression of this pair of observations in a 2D space that gives us least sum of residuals. We denote the linear regression by slope a_{ij} and intercept b_{ij}. Now the problem is to select a subset of feature so that the sum of residuals from the best regression is minimized. For the bi-cluster that is generated based on the ith observation, we introduce a $1 \times p$ feature selection vector $\mathbf{s_i} \in \{0, 1\}$, where $s_{ik} = 1$ if the kth feature is selected and 0 otherwise. Without any constraint, this problem will always give a trivial solution $\mathbf{s_i} = \mathbf{0}$, yielding a zero sum of residuals. Therefore, we add a regularizer $\beta_1 \|\mathbf{1} - \mathbf{s_i}\|_1$ to

penalize any solution with too few $s_{ik} = 1$ values, where $\mathbf{1}$ denotes a vector of all 1s and β_1 is the coefficient of the regularizer. In the subsequent formulations we choose the L_1 norm because it gives us a sparse solution in $\mathbf{1} - \mathbf{s_i}$ once $\mathbf{s_i}$ has been relaxed to $[0, 1]$. For a single row i, the problem can then be formulated as an optimization as follows:

$$\begin{aligned} \min_{\mathbf{s_i}, a_{ij}, b_{ij}} & \sum_k s_{ik}(m_{ik} - a_{ij}m_{jk} - b_{ij})^2 + \beta_1 \|\mathbf{1} - \mathbf{s_i}\|_1 \\ s.t. \quad & s_{ik} \in \{0, 1\} \end{aligned} \tag{1}$$

Now, consider the whole matrix M from which we want to detect a set of row-wise linear coherent bi-clusters. We introduce a $n \times n$ binary matrix W, where $w_{ij} = 1$ indicates there is strong linear coherence between the observation pair (i, j) and $w_{ij} = 0$ otherwise. By extending 1 in terms of the whole matrices M, S, A, B and introducing W, we obtain the complete formulation:

$$\begin{aligned} \min_{W,S,A,B} & \sum_{i,j} w_{ij} \sum_k s_{ik}(m_{ik} - a_{ij}m_{jk} - b_{ij})^2 \\ & + \beta_1 \|\mathbf{1} \cdot \mathbf{1}^T - S\|_{1,1} + \beta_2 \|\mathbf{1} \cdot \mathbf{1}^T - W\|_{1,1} \\ s.t. \quad & w_{ij} \in \{0, 1\}, s_{ik} \in \{0, 1\} \end{aligned} \tag{2}$$

where W can be interpreted as observation (data) selection matrix, and S can be interpreted as the feature (sample) selection matrix. β_2 is the coefficient of the W-wise regularizer. Here S is a $n \times p$ binary matrix with the ith row corresponding to the feature selection vector for the ith observation. Note that the sparse regularizer $\beta_1 \|\mathbf{1} - \mathbf{s_i}\|_1$ becomes $\beta_1 \|\mathbf{1} \cdot \mathbf{1}^T - S\|_{1,1}$. Similarly, we add another sparse regularizer $\beta_1 \|\mathbf{1} \cdot \mathbf{1}^T - W\|_{1,1}$ to penalize trivial solutions where W is set too close to the zero matrix.

We want to favor the case that the scatter points (feature points) of a pairwise 2D plot do not stick together so as to exhibit better linear coherence. Towards this end, we introduce a $n \times n \times p$ matrix D, where $d_{ijk} \in [0, 1]$ indicates the importance of the kth feature under the observation pair (i, j). In the gene expression matrix case, because it is desired to favor co-up-regulated and co-down-regulated gene expression samples, we assign $d_{ijk} = e^{d'_{ijk}}$, where d'_{ijk} is the Euclidean distance of the kth data point to the central point $(\bar{m}_{i:}, \bar{m}_{j:})$. Different prior knowledge can be introduced to form D from other data sources. Therefore, after relaxing $W \in \{0, 1\}$ to $W \in [0, 1]$ and $S \in \{0, 1\}$ to $S \in [0, 1]$, we get:

$$\begin{aligned} \min_{W,S,A,B} & \sum_{i,j} w_{ij} \sum_k s_{ik} \frac{1}{d_{ijk}} (m_{ik} - a_{ij}m_{jk} - b_{ij})^2 \\ & + \beta_1 \|\mathbf{1} \cdot \mathbf{1}^T - S\|_{1,1} + \beta_2 \|\mathbf{1} \cdot \mathbf{1}^T - W\|_{1,1} \\ s.t. \quad & w_{ij} \in [0, 1], s_{ik} \in [0, 1] \end{aligned} \tag{3}$$

By introducing some new notation, we can re-express this problem in an equivalent form that proves to be more convenient for formulating an efficient iterative procedure below. Let $\otimes$ denote Kronecker product, let $\triangle(\mathbf{m})$ denote putting a

vector $\mathbf{m}$ on the main diagonal of a square matrix, and let $\div$ denote component-wise division. Then 3 can be equivalently re-written in terms of $\mathbf{s_{i:}}$ and $\mathbf{w_{i:}}$ as:

$$\begin{aligned}\min_{W,S,A,B} \sum_i \Big\| \triangle(\mathbf{w_{i:}})^{1/2}(\mathbf{1} \otimes \mathbf{m_{i:}} - \triangle(\mathbf{a_{i:}})M - \triangle(\mathbf{b_{j:}})\mathbf{1} \otimes \mathbf{1}^T) \div D_i^* \triangle(\mathbf{s_{i:}})^{1/2} \Big\|_F^2 \\ +\beta_1 \|\mathbf{1} \cdot \mathbf{1}^T - S\|_{1,1} + \beta_2 \|\mathbf{1} \cdot \mathbf{1}^T - W\|_{1,1} \\ s.t. \quad w_{ij} \in [0,1], s_{ik} \in [0,1]\end{aligned} \tag{4}$$

where D_i^* has the same dimension as D_i with each element equal to the square root of the corresponding element in D_i.

Unfortunately, 4 is not jointly convex in W, S, A and B, so we are currently unable to formulate an efficient global optimization procedure. Nevertheless, an efficient iterative procedure can be devised that finds a reasonable local solution.

2.1 Initialization:

Because of the potential difficulty of local minima, initialization of W, S, A, and B becomes very important for solving 4 iteratively. To simplify the initialization, and allow a generally effective approach, we first normalize the data matrix M so that each row $\mathbf{m_{i:}} \in [0,1]$. In the case of gene expression analysis, A is initialized to $\mathbf{1} \cdot \mathbf{1}^T$ since a gene pair that has strong correlation will have a sufficient number of samples (features) under which the gene pair has a co-up-regulated and co-down-regulated pattern, which implies that on normalized data, the slope is near 1. The intercept b_{ij} is normalized in a way that the linear regression line for each observation pair passes through the central point $(\bar{m}_{i:}, \bar{m}_{j:})$ with slope a_{ij}. After A and B are initialized, $\mathbf{s_{i:}}$ is initialized such that $s_{ik} = 1$ if the distance d'_{ijk} of kth data point of the (i,j) pair to the line (a_{ij}, b_{ij}) is within some threshold. Since the data is normalized, an appropriate threshold can be set for data of the same type and will not affect the results to a large extent. In the case of gene expression data, since we want to favor sample points that are far away from the central point $(\bar{m}_{i:}, \bar{m}_{j:})$, we set the threshold as a monotonically increasing function of the distance d''_{ijk} between $(\bar{m}_{i:}, \bar{m}_{j:})$ and the projection of the (m_{ik}, m_{jk}) on the regression line. We do not initialize W as it will be immediately determined from the initial S, A, and B.

2.2 Iterative update of W, S, A and B:

Updating W. Denote the objective function in 4 by $f(W,S,A,B)$. Assume that S, A, and B are fixed (initialized as mentioned above for the first iteration). Then the objective function is a convex (linear) function of W and we can optimize W element by element in a closed form. In particular, for each w_{ij}, by ignoring constant terms, the problem is equivalent to minimizing $f(w_{ij})$:

$$\begin{aligned}\min_{w_{ij}} w_{ij} \left(\sum_k \frac{s_{ik}}{d_{ijk}} (m_{ik} - a_{ij} m_{jk} - b_{ij})^2 - \beta_2 \right) \\ s.t. \quad w_{ij} \in [0,1]\end{aligned} \tag{5}$$

Because $f(w_{ij})$ is a linear function of w_{ij}, we obtain $w_{ij} = 1$ if $\sum_k \frac{s_{ik}}{d_{ijk}}(m_{ik} - a_{ij}m_{jk} - b_{ij})^2 < \beta_2$ and $w_{ij} = 0$ otherwise.

Updating S. When W, A, and B are fixed, $f(W, S, A, B)$ becomes a convex (linear) function of S, so similar to updating W, we can update S element by element in a closed form. In this case, s_{ik} can be calculated by minimizing $f(s_{ik})$ as follows:

$$\begin{aligned} \min_{s_{ik}} s_{ik} & \left(\sum_j \frac{w_{ij}}{d_{ijk}}(m_{ik} - a_{ij}m_{jk} - b_{ij})^2 - \beta_1 \right) \\ s.t. \quad & s_{ik} \in [0, 1] \end{aligned} \tag{6}$$

Hence, $s_{ik} = 1$ if $\sum_j \frac{w_{ij}}{d_{ijk}}(m_{ik} - a_{ij}m_{jk} - b_{ij})^2 < \beta_1$ and $s_{ik} = 0$ otherwise.

Updating A and B. When W and S are fixed the minimization over A and B becomes a standard least squares linear regression problem for each observation pair. In particular, we have:

$$(a_{ij}, b_{ij})^T = (X_{ij}^T \triangle(\mathbf{s_{i:}} \bullet \mathbf{d_{ij:}}) X_{ij})^{-1} X_{ij}^T \triangle(\mathbf{s_{i:}} \bullet \mathbf{d_{ij:}}) \mathbf{y_{ij}} \tag{7}$$

where $\bullet$ denotes inner product, $X_{ij} = (\mathbf{1}, \mathbf{m_{j:}}^T)$, and $\mathbf{y} = \mathbf{m_{i,:}}^T$

Finally, each of W, S, A, and B are iteratively updated until the objective function converges. Algorithm 1 in Appendix gives the details of the SLLB algorithm. Note that the time complexity of the SLLB is $O(n^2)$ per iteration. Later experiments on synthetic datasets show that SLLB converges after 6-8 iterations, which takes less than 10 seconds in total. On real datasets, good results can be obtained after 10-20 iterations, which take tens of hours.

3 Results and Discussion

We compare the SLLB with seven existing representative bi-clustering algorithms, QUBIC, LinCoh, LCBD, CC, OPSM, ISA, and MSBE on synthetic datasets and two real gene expression microarray datasets on *Saccharomyces cerevisiae* (yeast) and *Escherichia coli* (e.coli) respectively. The parameter settings for the compared algorithms mostly follow the previous works [19, 15, 23].

3.1 Synthetic datasets

On synthetic datasets, Prelić's observation (gene) match score and overall match score [19] are adopted to evaluate the ability of bi-clustering algorithms in discovering the implanted (true) bi-clusters. 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 respectively. The observation 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 the average of the maximum observation match scores of bi-clusters in $\mathcal{C}$ with respect to the target bi-clusters. The feature match score $\text{score}_S(\mathcal{C}, \mathcal{C}^*)$ can be similarly defined

by replacing observation sets with the corresponding feature sets in the above. The overall match score is then defined as their geometric mean, *i.e.*

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

As for the parameter setting of SLLB, we set $\beta_1 = 0.1$, $\beta_2 = 0.05$ on all the noise resistance experiments, and we set $\beta_1 = 0.1$, $\beta_2 = 0.3$ on all the overlapping experiments.

Noise resistance test: This experiment investigates the ability of different bi-clustering algorithms in recovering implanted bi-clusters with different noise level. Following Prelić's testing strategy, we first generate a 100×50 background matrix based on a standard normal distribution and then embed ten 10×5 non-overlapping linear coherent bi-clusters along the diagonal. Then, for each vector of the five expression values, we set the first two to be down-regulated, the last two to be up-regulated, and the middle one to be non-regulated. Lastly, we add noise of six different levels ($\ell = 0.00, 0.05, 0.10, 0.15, 0.20, 0.25$) to the embedded bi-clusters by perturbing the entry values so that the resultant values are ℓ away from the original values. The generation is repeated ten times. Based on the same simulation process, we generate additive bi-clusters on synthetic datasets when we compare the bi-clustering algorithms on their performance in discovering additive bi-clusters only (which is a special case of linear coherent bi-clusters).

Figure 6 in Appendix shows the observation match scores of the bi-clusters discovered by the eight algorithms at six different noise levels. Figures 7 and 8 in the Appendix demonstrate their overall match scores and observation discovery rates (defined as the percentage of observations in the output bi-clusters over all the observations in the true bi-clusters). From these figures, it is clearly shown that SLLB outperformes all the other seven algorithms; QUBIC, LinCoh and ISA rank the second, third, and forth, and the other three performed quite poorly. Note that by simply outputting more bi-clusters, observation discovery rate can be trivially lifted up. Therefore it is only a useful measurement in conjunction with match scores.

Overlapping test: Bi-clusters may overlap in terms of either observations or features. Take gene expression as an example, some genes can participate in multiple biological processes which result in bi-clusters that overlap with common genes in an expression matrix. It is also the case in sample overlapping. This experiment intends to examine the ability of bi-clustering algorithms in recovering overlapping bi-clusters. We again consider type-(f) linear coherent bi-clusters and type-(d) additive bi-clusters, at a fixed noise level of $\ell = 0.1$. We generate ten 100×50 matrices based on a standard normal distribution. In each matrix, two 10×10 bi-clusters are embedded, with overlapping size: 0×0, 1×1, 2×2, 3×3, 4×4, and 5×5. In the case of gene expression, we assume that these overlapping genes obey a reasonable logic such as the AND gate and the OR

gate which leads to a behavior of *union* and an *additive* respectively. So the overlapped entries in the union overlapping area preserve linear coherency in both bi-clusters and in the additive overlap model, these entries are assigned by the sum of the gene expression levels from both bi-clusters. The observation match

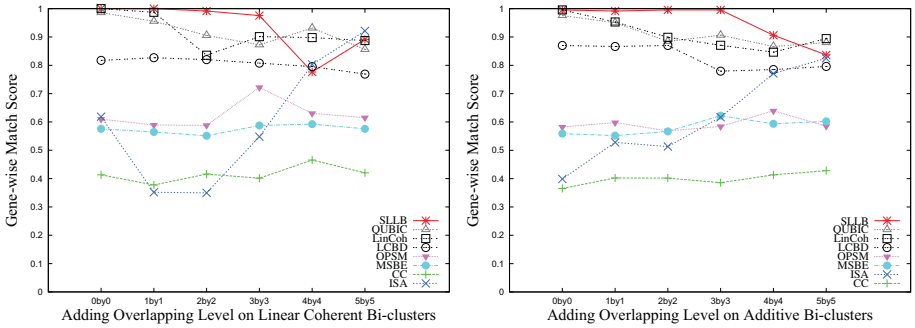

Fig. 1. The observation match scores of the eight algorithms for recovering the overlapping linear coherent and additive bi-clusters, under the adding overlap model.

scores of the eight bi-clustering algorithms in this adding overlapping experiment are shown in Figure 1. Figures 9 and 10 in the Appendix plot the overall match scores and observation discovery rates under the adding overlap model. The results of the additive overlap model are shown in Figures 11, 12, and 13 in the Appendix. From all these results, we can conclude that SLLB outperformes the other seven algorithms. LinCoh's performance is slightly worse than SLLB; QUBIC, OPSM and MSBE perform worse, but similarly to each other; LCBD and CC performed the worst; and ISA demonstrates varying performance.

3.2 Real datasets

On real datasets, the quality of bi-clusters is evaluated by known biological pathways, defined in the GO functional classification scheme [1], the KEGG pathways [12], the MIPS yeast functional categories [20] (for yeast dataset), and the EcoCyc database [13] (for e.coli dataset), in order to obtain their *gene functional enrichment score* as implemented in [14]. The average correlation coefficient is also used for evaluating the generated bi-clusters on real datasets.

We obtain the yeast dataset from [8]. It contains 2993 genes on 173 samples; the e.coli dataset is obtained from [6], (version 4 built 3). It contains initially 4217 genes on 264 samples. For the e.coli dataset, after removing genes with too small expression deviations, we get 3016 genes. This pre-process ensures that all eight bi-clustering algorithms can be run on the dataset. We use the gene functional enrichment score [14] to measure the performance of different algorithms. 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 [14],

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

Such a probability is taken as the P-value of the output bi-cluster enriched with genes from that functional class [14]. The P-value of the output bi-cluster is defined as the smallest P-value over all functional classes. The smaller the P-value of a bi-cluster the more likely do its genes come from the same biological process. We calculate for each algorithm the fraction of its output bi-clusters whose P-values are smaller than a significance cutoff α. As for the parameter setting of SLLB, we set $\beta_1 = 0.3$, $\beta_2 = 1.5$ for the yeast dataset, and $\beta_1 = 0.1$, $\beta_2 = 0.5$ for the e.coli dataset.

In Figure 2 the eight algorithms are compared using six different P-value cutoffs, evaluated on the GO database. Results on the KEGG, MIPS, and Regulon databases are in Figures 14 and 15 in Appendix. These results indicates that SLLB performs consistently well; QUBIC and LinCoh performs stable but worse than SLLB, OPSM and ISA does not perform consistently on the two datasets across databases; and that LCBD, MSBE and CC does not perform as well as the other three algorithms.

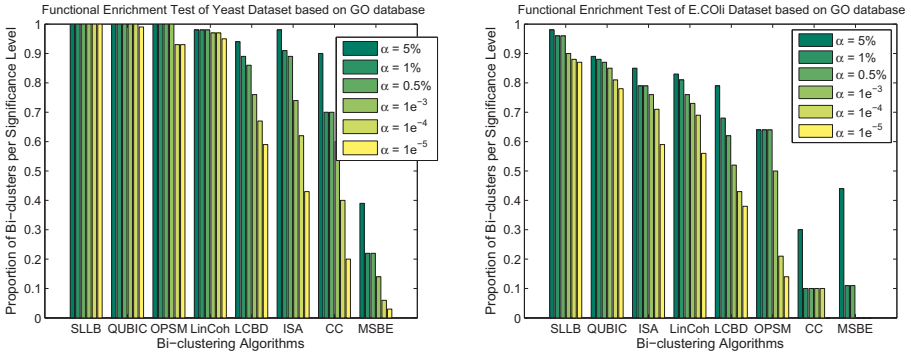

Fig. 2. Portions of discovered bi-clusters by the eight algorithms on the two real datasets that are significantly enriched in the GO biological process, using six different P-value cutoffs.

One potential issue with the P-value based performance measurement is that P-values are sensitive to the bi-cluster size [14]; in general, this measurement favors bi-clusters with a larger size. For example, in Table 1, it is shown that OPSM finds bi-clusters that contain extremely large number of genes and very

few samples. Bi-clusters of this kind are close to trivial bi-clusters (gene or sample set size close to 0) but with large number of gene, its enrichment P value can be easily lifted up. On the contrary, although our SLLB algorithm generates bi-clusters with large number of genes, the number of samples it generates is also large which indicates more confident linear coherence. In the last column of the Table 1, the numbers of unique functional terms enriched by the produced bi-clusters are listed. When measured by the gene enrichment significance score, OPSM performed very well on yeast dataset (Figure 2, left), but its bi-clusters only cover one functional term on the GO and KEGG databases and two terms on MIPS database. This suggests that the bi-clustering result can be biased to a group of correlated genes, which are missed by the P-value based significance test.

Considering these two potential issues, we can see that the P-value based evaluation is meaningful but has limitations. So we propose using the average absolute correlation coefficient over all gene pairs in a bi-cluster as an alternative assessment of the quality of a linear coherent bi-cluster. However, note that the numbers of samples in the bi-clusters generated by some algorithms are much smaller others, Table 1. Therefore, 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 [21, 23]. These values are plotted in Figure 3.

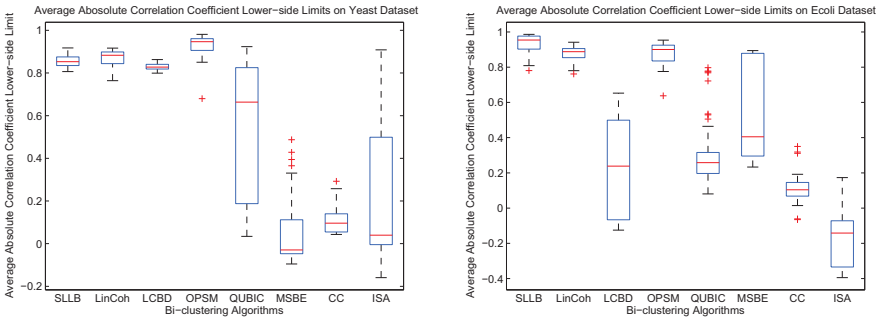

Fig. 3. Box plots of the average absolute correlation coefficients obtained by the eight bi-clustering algorithms on yeast and e.coli datasets, respectively.

Figure 3 shows that SLLB, LinCoh, and OPSM have similarly good performance while QUBIC, LCBD, MSBE, CC, and ISA performs worse than these three. Note that due to noise effect when profiling genes, itt is hard to reach a very large value of the correlation coefficient.

4 Conclusion

In this article, we proposed a novel bi-clustering algorithm, SLLB, that can discover linear coherent bi-clusters based on a sparse learning optimization model. The experimental results on both synthetic and real datasets indicate that SLLB is not only able to discover linear coherent bi-clusters effectively, but able to discover meaningful linear coherent bi-clusters that can be verified by biological ground truth. Actually, for many bi-clusters discovered by SLLB, all their corresponding gene groups (with size 30-100) belong amazingly to the same gene ontology term. The time complexity of the SLLB algorithm is $O(n^2k)$ where n is the number of observations and k is the number of iterations that SLLB takes to converge, which is very fast compared to algorithms like LinCoh. Note that while discovering linear coherent bi-clusters, SLLB favors data points corresponding to features that are far away from each other in the observation pair 2D space. This nice property can be used for downstream data analysis such as feature clustering, observation-feature relation studies and observation/feature selection.

To set appropriate values for β_1 and β_2, We binary searched $\beta_1 \in [0, 1000]$ and $\beta_2 \in [0, 1000]$ and found the value ranges that produce non-trivial bi-clusters are $\beta_1 \in [0, 0.5]$ and $\beta_2 \in [0, 1.5]$. We then tested different combinations of $\beta_1 = [0.1, 0.5, 1]$ and $\beta_2 = [0.1, 0.5, 1, 1.5]$ and found the results are quite robust to different settings. The final β_1 and β_2 are chosen so that SLLB performs best. When come to practice, considering that β_1 actually controls the size of observation and β_2 controls the size of features in the result bi-clusters, β_1 and β_2 can be determined when prior knowledge of bi-cluster size is known.

We suggest that the SLLB algorithm can be used in other machine learning applications such as image clustering, document clustering, and other biology and health care data clustering, as long as observations of the same group have linear coherence under a subset of features, and for different clusters, different feature sets need to be selected.

As for future work, we will test SLLB on other applications such as document bi-clustering and image bi-clustering. We will also extend SLLB to consider other relations between observations in addition to the linear coherent relations.

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. W. Ayadi, M. Elloumi, and J. K. Hao. Pattern-driven neighborhood search for biclustering of microarray data. *BMC Bioinformatics*, 13:(Suppl 7):S11, 2012.
3. 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.
4. Y. Cheng and G. M. Church. Biclustering of expression data. In *ISMB'00*, 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. *PNAS*, 95:14863–14868, 1998.

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. R. Gupta and V. Kumar N. Rao. Discovery of error-tolerant biclusters from noisy gene expression data. *Bioinformatics*, 12:(Suppl 12):S1, 2011.
10. J. A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67:123–129, 1972.
11. J. Ihmels, S. Bergmann, and N. Barkai. Defining transcription modules using large scale gene expression data. *Bioinformatics*, 20:1993–2003, 2004.
12. M. Kanehisa. The KEGG database. *Novartis Foundation Symposium*, 247:91–101, 2002.
13. 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.
14. 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.
15. X. Liu and L. Wang. Computing the maximum similarity bi-clusters of gene expression data. *Bioinformatics*, 23:50–56, 2006.
16. 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.
17. J. Meng and Y. Huang. Biclustering of time series microarray data. *Methods Mol Biol*, 802:87–100, 2012.
18. B. Mirkin. Mathematical classification and clustering. *Kluwer Academic Publishers*, 1996.
19. 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.
20. 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.
21. D. Shen and Z. Lu. Computation of correlation coefficient and its confidence interval in SAS. http://www2.sas.com/proceedings/sugi31/170-31.pdf.
22. Y. Shi, Z. Cai, G. Lin, and D. Schuurmans. Linear coherent bi-cluster discovery via line detection and sample majority voting. In *COCOA'09*, pages 73–84, 2009.
23. Y. Shi, M. Hasan, Zhipeng Cai, G. Lin, and D. Schuurmans. Linear coherent bi-cluster discovery via beam detection and sample set clustering. *International Conference on Combinatorial Optimization and Applications*, 1:85–103, 2010.
24. 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.
25. 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.