JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 19, Number 6, 2012

Pp. 619–631
DOI: 10.1089/cmb.2012.0032

Biclustering of Linear Patterns In Gene Expression Data

QINGHUI GAO,[1] CHRISTINE HO,[3] YINGMIN JIA,[1,2]
JINGYI JESSICA LI,[3] and HAIYAN HUANG[3]

ABSTRACT

Identifying a bicluster, or submatrix of a gene expression dataset wherein the genes express similar behavior over the columns, is useful for discovering novel functional gene interactions. In this article, we introduce a new algorithm for finding biClusters with Linear Patterns (CLiP). Instead of solely maximizing Pearson correlation, we introduce a fitness function that also considers the correlation of complementary genes and conditions. This eliminates the need for *a priori* determination of the bicluster size. We employ both greedy search and the genetic algorithm in optimization, incorporating resampling for more robust discovery. When applied to both real and simulation datasets, our results show that CLiP is superior to existing methods. In analyzing RNA-seq fly and worm time-course data from modENCODE, we uncover a set of similarly expressed genes suggesting maternal dependence. Supplementary Material is available online (at www.liebertonline.com/cmb).

Key words: algorithms, gene clusters, probability.

1. INTRODUCTION

Traditional clustering methods, such as hierarchical clustering (Johnson, 1967), K-means clustering (Hartigan, 1972), self-organizing maps (Tamaryo et al., 1999), and model-based methods (Banfield and Raftery, 1993; Ben-Dor et al., 2003; Fraley and Raftery, 2002; Mclachian and Basford, 1998) can organize gene expression data into clusters of genes possessing similar expression profiles over the whole set of given experimental conditions. However, the intrinsic complexity of gene expression data, especially when the experimental conditions are diverse, suggests that identifying groups of genes exhibiting local, rather than global, association patterns is a better strategy for obtaining biologically relevant results. For this purpose, biclustering is a useful data-mining technique, involving the simultaneous clustering of genes and experimental conditions in a gene expression matrix. This allows the discovery of subsets of genes that are co-regulated or co-expressed only under certain experimental conditions.

It is worthwhile to note that the term ''biclustering'' has been used in the literature to refer to very different ideas. Methods range from requiring a complete partition of the data matrix, wherein the resultant biclusters must be distinct in both the gene and condition dimensions, to imposing no constraints on the number of genes, conditions, or the degree of overlap in the biclusters. This is discussed in greater detail in the literature review by Madeira and Oliveira (2004).

[1]Seventh Research Division and Department of Systems and Control, Beihang University, Beijing China.
[2]Key Laboratory of Informatics Mathematics and Behavioral Semantics, SMSS, Beihang University, Beijing, China.
[3]Department of Statistics, University of California, Berkeley, California.

The performance of a biclustering algorithm hinges upon the choice of a similarity measure and fitness function, as well as the way in which optimization of the fitness function is implemented. With regard to the former, the similarity measure and fitness function define the *kind* of local pattern of interest. Existing algorithms can capture a constant pattern ($b_{ij} = \mu$, where b_{ij} is an element of the bicluster submatrix of genes and conditions B) (Hartigan, 1975; Busygin et al., 2002); the additive model ($b_{ij} = \mu + \alpha_i + c_j$) (Getz et al., 2000; Califano et al., 2004; Sheng et al., 2003); and the multiplicative model ($b_{ij} = \mu \times \alpha_i \times c_j$ or equivalently $B = \alpha \cdot c^T$, where α and c are column vectors) (Cheng and Church, 2000; Yang et al., 2003; Wang et al., 2002; Lazzeroni and Owen, 2002; Segal et al., 2003; Tang et al., 2001; Klugar et al., 2003; Hochreiter et al., 2010). The term coherent model has been used in the literature to describe biclusters with either additive or multiplicative patterns, to distinguish it from the more inflexible constant model (Madeira and Oliveira, 2004). Another common pattern is the model of coherent evolutions, which has also been referred to in some papers as a scaling pattern (Li et al., 2009). We caution that the term ''scaling pattern'' is ambiguous, and can also refer to the multiplicative model (Aguilar-Ruiz, 2005); for clarity, we adhere to the term ''coherent evolutions.'' This model describes a consistent linear ordering of expression values across the conditions, and can therefore capture patterns involving both up- and down-regulated genes (Ben-Dor et al., 2003; Tang et al., 2001; Murali and Kasif, 2003; Liu and Wang, 2003; Mclachian and Basford, 1998; Li et al., 2009).

A more general variant of the constant and coherent models is the linear model, where $b_{ij} = \mu_i + \alpha_i c_j$ (equivalently, $B = \alpha \cdot c^T + \mu 1$ with 1 the matrix of all ones and $\mu = Diag(\mu_1, \ldots, \mu_d)$ a diagonal matrix). In this model, any two rows (genes) in the bicluster hold an exact linear relationship. The linear model generalizes the previous models: all but one of them are specific versions of this model, the exception being the loosely defined coherent evolution model. The Pearson correlation coefficient is commonly used as a similarity measure for capturing linear relationships, and is grandly useful for its generality. However, with noisy data, using Pearson correlation can result in a large number of false positives, since it only assesses the profile shape and not the profile elevation. We note that the present literature on biclustering of linear patterns is limited. An extensive literature search only found BCCA (Bhattacharya and De, 2009), Scatter Search (Nepomuceno et al., 2011), and geometric biclustering algorithms based on the fast Hough transform (Gan et al., 2005, 2008; Zhao et al., 2008). Literature on the other aforementioned models is richer, as just referenced.

Given an optimality criterion for the pattern of interest, such as Pearson correlation, biclustering algorithms differ widely in how they implement optimization. Due to the combinatorial nature of the problem, enumerative methods are computationally impractical. In fact, at its simplest, Madeira and Oliveira (2004) have shown that the biclustering problem is NP-complete. As a result, existing methods propose means of finding local, rather than global optima. Methods include greedy search, top-down procedures, iterative combination of one-way clustering results, and in model-based techniques, variations on the Expectation-Maximization algorithm. Again, we refer to Madeira and Oliveira (2004) for a more thorough discussion of these techniques.

This article introduces a new method, CLiP, for biClustering of Linear Patterns in gene expression data. The biclusters returned by the method are flexible in form: they may share genes or conditions, and do not represent a strict partition of the original matrix. We define similarity measures based on the Pearson correlation, and include a variance term to exclude lowly expressed background noise. The strength of the method lies in the design of the fitness function, which compares the association in a bicluster with that within its complementary sets, a concept we refer to as ''contrast.'' To maximize the fitness function, we devise a multi-step procedure that incorporates resampling for more robust discovery, and combines both evolutionary and greedy approaches. To reduce redundancy in the results, we introduce a seed-selection step similar in spirit to those proposed in Tanay et al. (2002) and Ihmels et al. (2002).

As mentioned earlier, directly competing methods for identifying linear patterns are BCCA (Bhattacharya and De, 2009), Scatter Search (Nepomuceno et al., 2011), and the geometric algorithms (Gan et al., 2005, 2008; Zhao et al., 2008). BCCA uses purely greedy search to identify biclusters in which pairwise correlation between genes meets a predefined threshold. Scatter Search employs an evolutionary algorithm to identify biclusters with the highest average pairwise correlation by gene. The geometric algorithms cast a bicluster as a hyperplane with a specific set of linear geometries, and optimize using the fast Hough transform-based hyperplane-detection algorithm of Li et al. (1986). Although the geometric algorithms present a new perspective on biclustering of linear patterns, they capture the same relationship as Pearson

correlation. Further, these algorithms suffer from poor scalability and do not possess advantages over the other two methods, which appeared later in the literature.

We tested the performance of CLiP on simulation and real data sets, and our results indicate that CLiP shows significant improvement over BCCA in identifying biclusters with a local linear pattern. Our method is also comparable with, if not better than, QUBIC (Li et al., 2009), FABIA (Hochreiter et al., 2010), ISA (Mclachian and Basford, 1998; Ihmels et al., 2004), and BCCA in identifying additive and scaling patterns. An indirect comparison suggests that CLiP is also superior to Scatter Search and the geometric algorithm (both methods do not provide publicly available code). Source code for CLiP is available upon request.

2. METHODS

We split discussion of our methodology into two parts. In Section 2.1, we motivate and define our similarity measures and fitness function. In Section 2.2, we describe our implementation for optimizing the fitness function, as well as a procedure for eliminating redundant biclusters in the result.

2.1. Similarity measure and objective function.

Let X be an $n \times m$ matrix of expression values for n genes and m experimental conditions. Denote by $B(R, C)$ the submatrix given by the gene and condition subsets R and C, respectively.

2.1.1. Similarity measure. A similarity measure assesses the strength of the pattern of interest in a submatrix as well as the relatedness of two different submatrices. Because we are interested in linear relationships, a natural choice is to construct a similarity measure using the Pearson correlation coefficient. As such, we introduce correlation-based similarity measures $r_C(\cdot,\cdot)$ and $r_R(\cdot,\cdot)$ for describing the linear association along the gene and condition dimensions, respectively. Since correlation only evaluates the shape of the data, we also include a measure of variance, $d(\cdot,\cdot)$, for excluding lowly expressed background noise that may produce high correlations by chance. For a submatrix $B(R, C)$, the Pearson correlation coefficient between two genes $g_i = (x_{i1}, \ldots, x_{im})$ and $g_j = (x_{j1}, \ldots, x_{jm})$ is defined as

$$corr_C(g_i, g_j) = \frac{\sum_{k\in C}(x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)}{\sqrt{\sum_{k\in C}(x_{ik} - x_i)^2}\sqrt{\sum_{k\in C}(x_{jk} - x_j)^2}},$$

where $\bar{x}_i$ denotes the average expression value of gene g_i over the conditions of C. Similarly, let $corr_R(\cdot,\cdot)$ denote the correlation between two conditions, holding the set of genes fixed.

Generalizing from pairs of genes, we define the following similarity measures $r_C(\cdot,\cdot)$ and $r_R(\cdot,\cdot)$ for submatrices in two parts: first, for comparing within a submatrix; secondly, for comparing two submatrices.

For measuring linearity within a submatrix, we define $r_C(\cdot,\cdot)$ to be the average of all pairwise correlations between a gene and the mean vector calculated without that gene, i.e., a ''leave-one-out'' mean. The use of a ''leave-one-out'' mean is motivated by experience; in practice, on submatrices with few rows, including the gene under consideration in computing the mean tends to produce artificially large correlations. A measure for correlation along the condition dimension is defined in a similar way. To be precise,

$$r_C(R, R) = \frac{1}{|R|}\sum_{g_i\in R} corr_C(g_i, \bar{g}_R^{(i)})$$

$$r_R(C, C) = \frac{1}{|C|}\sum_{c_i\in C} corr_R(c_i, \bar{c}_C^{(i)}),$$

where $\bar{g}_R^{(i)}$ is the vector obtained by taking the mean across genes in $R \setminus \{g_i\}$, and $\bar{c}_C^{(i)}$ is similarly defined. One might notice the use of signed correlation in these measures. Because our method aims to find consistent patterns of expression, we penalize cases of negative correlation, which indicate an inverse regulatory relationship.

For the similarity between two different submatrices, we define a natural extension of the prior two measures, using the same notation as before. Holding the conditions C fixed, let $r_C(Z, R)$ denote the average correlation between genes in Z to the mean vector of R; if this is high, then many genes in Z exhibit the

same relationship as genes in R. Similarly, $r_R(Y, C)$ describes the similarity between condition sets Y and Z, holding genes fixed. Formally,

$$r_C(Z, R) = \frac{1}{|Z|} \sum_{g_i \in Z} corr_C(g_i, \bar{g}_R)$$

$$r_R(Y, C) = \frac{1}{|Y|} \sum_{g_i \in Y} corr_R(c_i, \bar{c}_C).$$

As alluded to earlier, using Pearson correlation alone on noisy data can result in false positives. As an extreme example, the Pearson correlation between (0.1, 0.3, 0.1, 0.2) and (100, 300, 100, 200) is 1, but these two profiles are very different. The former profile represents random background noise, while the latter represents a gene with significant expression changes across experiments. When the number of conditions under consideration is small, as is typical for gene expression data, the chance that background variation exhibits a linear pattern similar to a real bicluster is non-negligible. This motivates the inclusion of a measure that quantifies the variance in expression levels within a bicluster, so as to exclude this highly correlated low-expression noise.

Recall the definition of Euclidean distance, here taken between two genes:

$$norm_C(g_i, g_j) = \sqrt{\sum_{k \in C} (x_{ik} - x_{jk})^2}$$

Below, we define our measure of variance within a bicluster, $d(\cdot, \cdot)$, as the average distance to the leave-one-out mean gene vector. This is similar to the standard deviation of gene expression values within the submatrix.

$$d(R, C) = \frac{1}{|R|} \sum_{g_i \in R} norm_C(g_i, \bar{g}_{(i)}).$$

2.1.2. Objective function. An objective or fitness function evaluates the optimality of a bicluster with respect to the pattern of interest, and thus depends on the choice of similarity measure. As the example with background noise suggests, attaining maximal correlation may not be the best optimality criterion in searching for linear patterns. More critically, a serious weakness of methods that solely maximize correlation is that they fail to include genes that exhibit the linear pattern less strongly than others, but nonetheless belong in the bicluster. In other words, these genes may be strongly associated with those of the bicluster, but not strongly enough to be included when maximizing correlation alone. This becomes especially problematic with methods requiring a predefined correlation threshold, such as BCCA: all genes failing to meet the threshold are excluded from the bicluster. As a solution to this problem, we introduce a fitness function that favors high correlation within a bicluster and penalizes occurrence of the bicluster pattern in complementary genes; essentially, we reward high ''contrast'' between a bicluster and its complement. The same argument can be made for the condition dimension, prompting a similar penalty.

The fitness function is comprised of four additive components: f_c, which characterizes contrast along the gene dimension, holding conditions fixed; f_r, for contrast along the condition dimension, holding rows fixed; f, for contrast with the submatrix given by both the complement set of genes and conditions; and h, a term based on the variance measure of the previous section. Tying these components together, we define the fitness function $\Psi(\cdot, \cdot)$ as

$$\Psi(R, C) = f_r(R, C) + f_c(R, C) + f(R, C) + h(R, C).$$

In the paragraphs that follow, we describe each of these four components in turn.

Consider firstly the problem of maximizing contrast along the gene dimension. For the ideal submatrix $B(R, C)$, the correlation between genes in R on the conditions C should be maximal; at the same time, the correlation between the genes of R and R^- on the condition set C should be minimal. For the former, we use $r_C(R, R) + \alpha r_R(C, C)$ as a measure of association within the bicluster (α is explained below); for the latter, $r_C(R^-, R)$ is an intuitive choice. We base a measure of fitness along the gene dimension, $f_r(\cdot, \cdot)$, on these two measures, defining it as follows.

$$f_r(R, C) = \frac{|C|}{m}(r_C(R, R) + \alpha\ r_R(C, C)) - \frac{|C-|}{m}\frac{2}{\pi}\arctan(k_1\ r_C(R-, R)) + \frac{|C-|}{m}\theta_c$$

The multiplicative weights $\frac{|C|}{m}$ and $\frac{|C-|}{m}$, as well as the additive term $\frac{|C-|}{m}\theta_c$, penalize the size of the bicluster. In particular, larger values of θ_c penalize size more heavily by requiring more substantial gains in correlation to add conditions. As such, θ_c can be regarded as a tuning parameter for tightness, eliminating the need for *a priori* determination of the size of the bicluster. The tuning parameter α is binary; we set $\alpha = 0$ when the interpretability of the condition subset is less important or the pattern of interest deviates somewhat from a linear pattern. The arctan transformation and tuning parameter k_1 amplify the value of $r_C(R^-, R)$ around zero, allowing for greater sensitivity to changes in $r_C(R^-, R)$; these changes tend to be small due to the large size of the complementary set of genes, R^-.

For characterizing contrast along the condition dimension, we define f_c similarly.

$$f_c(R, C) = \frac{|R|}{n}(r_C(R, R) + \alpha\ r_R(C, C)) - \frac{|R^-|}{n}\frac{2}{\pi}\arctan(k_1\ r_R(C^-, C)) + \frac{|R^-|}{n}\theta_r$$

Further, we introduce a function that weighs the association within $B(R, C)$ against that of $B(R^-, C^-)$, the submatrix given by both complementary genes and conditions.

$$f(R, C) = \frac{1}{2}\left(\frac{|R|}{n} + \frac{|C|}{m}\right)(r_C(R, R) + \alpha\ r_R(C, C)) - \frac{1}{2}\left(\frac{|R^-|}{n} + \frac{|C^-|}{m}\right)\frac{2}{\pi}\arctan(k_1 r_{C-}(R^-, R^-)) + \frac{1}{2}\left(\frac{|R^-|}{n} + \frac{|C^-|}{m}\right)\theta$$

Finally, we define a function for penalizing high variance within a bicluster, to avoid situations like the background noise example of the previous section. Using the arctan transformation to restrict the range of the measure $d(\cdot, \cdot)$, we arrive at the following definition for h.

$$h(R, C) = \lambda\frac{|R|}{n}\left[1 - \frac{2}{\pi}\arctan(k_2\ d(R, C))\right]$$

Above, λ is a tuning parameter that controls the extent to which variance is penalized, while k_2 controls sensitivity to changes in variance. As an extreme example, setting λ and k_2 to very large and small values, respectively, will favor biclusters with nearly constant patterns.

2.2. *Optimization.*

Upon defining the fitness function, we have reduced finding a bicluster with linear pattern to a combinatorial optimization problem. Because we are considering all possible submatrices of a high-dimensional matrix, the search space is vast. Existing heuristic search algorithms balance computational efficiency with better

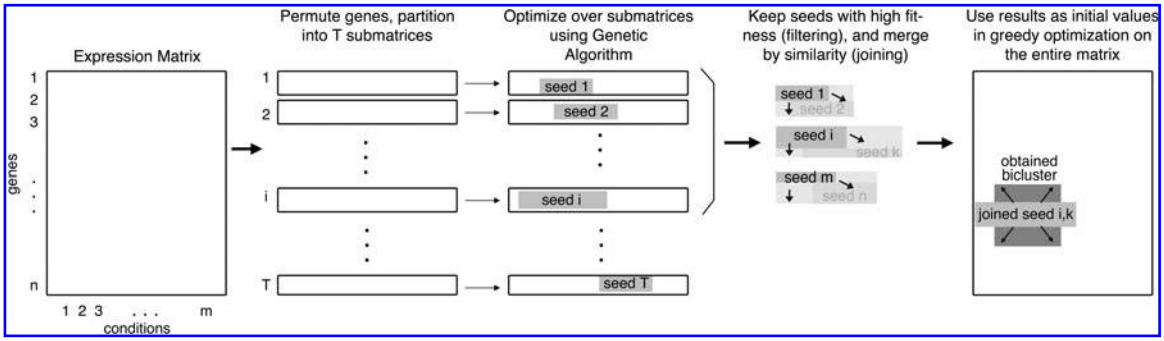

FIG. 1. A diagram of one iteration of the optimization procedure used in CLiP. In the first step, the data is randomly partitioned into submatrices of roughly equal size. Stochastic optimization using GA is performed on each of these submatrices to produce seeds. After filtering and joining, the seeds are used as starting values for greedy optimization on the whole expression matrix.

approximations to global optima. Hill climbing, here equivalent to greedy one-at-a-time modification of the rows and conditions, is extremely efficient but deterministic, thereby depending heavily on the initialization. In contrast, evolutionary algorithms like the genetic algorithm (GA) are stochastic and capable of multi-directional search. As such, the GA is well-suited for quickly locating a good, though likely suboptimal, solution in a large search space. However, defining appropriate stopping criteria for the GA is tricky, especially since the algorithm can become stuck at local optima for long periods of time. Further, unlike greedy methods, the GA can be computationally intensive.

To take advantage of the qualities of both approaches, we propose a multi-step optimization procedure that uses the GA to identify good starting seeds for greedy optimization (Fig. 1). The steps of the procedure are as follows: randomly partitioning the matrix by row into T submatrices; GA optimization on each of the submatrices to produce T starting seeds; filtering and combining ''similar'' starting seeds to reduce redundancy; greedy optimization, initializing with the seeds resulting from the previous step. We discuss each of these steps in greater detail below.

2.2.1. Optimization on random partitions. The procedure begins with randomly partitioning the matrix into T equally sized (or as close to equally sized as possible) submatrices possessing the full set of conditions; to be clear, each submatrix has at least $\lfloor n/T \rfloor$ rows, and m columns. This is done t times, so that this step yields $t \times T$ total submatrices. We refer to the value of the tuning parameter t as the number of iterations.

Then, GA optimization of the fitness function is performed on each of these submatrices, using as stopping criteria a maximum number of generations. In this application of the genetic algorithm, the ''individuals'' are bitstrings of length at least $\lfloor n/T \rfloor + m$, indicating set containment. We constrain the solution space so that the number of rows and conditions cannot fall below predefined thresholds. At minimum, a submatrix must have at least two rows and three columns for correlation to be well-defined. In addition, we leave as tuning parameters the number of individuals in each generation, a.k.a. the population size, and the total number of generations run. In general, initializing the GA with a larger population size results in a more exhaustive search of the solution space, at the expense of computational efficiency.

Note that the choice of the number of partitions T directly affects the efficiency of the algorithm. For an analysis of this dependence on T, see Supplementary Material (available online at www.liebertonline.com/cmb).

2.2.2. Filtering and joining of seed biclusters. When the number of partitions T is set to a high value, many seeds found in the GA optimization may be uninformative. To shave computational time by reducing the number of seeds passed to later steps, we rank them by their fitness value on the whole matrix, and retain the top *num_seeds* results, where *num_seeds* $\leq T$ is a predefined amount. We refer to this as our filtering step.

Next, we merge similar seeds to avoid redundancy in the final results. This redundancy occurs because true biclusters may be split in the initial partitioning of the matrix, causing many seeds to express the same pattern. If used as initial values in the final greedy optimization, these tend to grow into essentially the same bicluster. The procedure we adopt is similar to those of Tanay et al. (2002) and Ihmels et al. (2002). A description of our approach is provided in Supplementary Material.

2.2.3. Optimization over the whole dataset. Using the $\tilde{T} \leq T$ seed biclusters from the previous step as initial values, greedy optimization proceeds by iteratively adding genes, then conditions, one at a time until the fitness function can no longer be increased.

For large datasets, this step is the main bottleneck in efficiency, requiring more time than GA optimization on submatrices. This occurs because the number of fitness function evaluations per iteration is directly proportional to matrix size; whereas for GA, that number depends only on the predefined initial population size. The seed filtering and joining procedures of the previous section improve runtime in this step by cutting down on the number of seeds, and thus the number of independent greedy searches performed. A more rigorous analysis of the time complexity for these steps appears in Supplementary Material.

3. RESULTS

We tested the performance of CLiP on both simulation and real data sets, and compared our results to those of BCCA and several popular biclustering methods in the literature. For the other two methods that capture linear patterns, Scatter Search and the geometric algorithm, source code was not publicly available.

Table 1. Default Settings of the Tuning Parameters for CLiP

Minimum size	5×5
Fitness function	$t = 1$, $\theta_c = \theta_r = \theta = 1$, $\alpha = 1$, $k_1 = 1$, $k_2 = 1$, $\lambda = 2$
Genetic algorithm	Initial population size = 100, total generations = 200
Seed selection	Similarity threshold for rows, columns = 0.6

An indirect comparison based on their published results shows that CLiP performs just as well, if not better. The details of this comparison are given in the Supplementary Materials.

This section is organized in the following way. We first compare against BCCA on simulation data embedded with biclusters following a linear pattern. We then evaluate CLiP's performance in identifying additive and scaling patterns on simulation data, and present comparisons with other methods designed to capture these patterns. Next, we assess CLiP's ability to identify biologically relevant relationships on five well-studied datasets, and present a comparison with both BCCA and other methods. Finally, we present interesting findings from applying to CLiP to fly and worm time-course RNA-seq data generated by the modENCODE consortium.

For the analyses that follow, we use the default settings of the tuning parameters listed in Table 1 unless otherwise specified.

3.1. Simulation data

3.1.1. Biclusters with linear patterns. For validation, we generate three very different simulation datasets, each with two embedded non-overlapping linear biclusters. In the first data set, we consider an ideal, low-noise setting. We embed two 10×10 linear biclusters with mean expression values 130 and 85, respectively, in a matrix of size 200×150, allowing all other values to be 0. Then, we add mean-zero Gaussian noise, with standard deviation ranging from 0 to 0.25. For the second dataset, we consider a more realistic setting. We use the same biclusters as the previous dataset, but instead generate the background expression values from a Uniform distribution on $\{0, \ldots, 40\}$. For the third and most realistic dataset, we directly sample the distribution of expression values from the yeast cell cycle dataset of Tavazoie et al. (1999), consisting of microarray expression data for 2884 genes over 17 experiments. To create a bicluster, a length 10 vector c from the real dataset is drawn, and another length 10 vector α is generated with values randomly drawn from (0, 2]. The resultant bicluster is obtained by rounding the values of αc^T . These biclusters had average expression value of 5.8 and 6.4, in a matrix where the overall average expression value was 5.5.

To compare the results of CLiP and BCCA on these datasets, we use the average module recovery score of Prelic et al. (2006), which has been commonly used in the literature to describe the extent to which the obtained biclusters match the true ones. This score is defined as

$$S(M, M_d) = \frac{1}{|M|} \sum_{G \in M} \max_{G_d \in M_d} \frac{|G_d \cap G|}{|G_d \cup G|},$$

where M is the set of true biclusters (here, $|M| = 2$), M_d the set of obtained biclusters, and G (G_d) is the set of genes for the implanted (recovered) bicluster. In our evaluation, we restrict $|M_d| = |M|$, using only the best $|M|$ obtained biclusters. Intuitively, the score measures the extent to which the top $|M|$ biclusters match the true $|M|$ biclusters along the gene dimension. A perfect match yields a recovery score of 1.

For the first two datasets, a single iteration of CLiP was run using $T = 10$ partitions, an initial GA population size of 100 and number of generations of 200. No seed filtering was done, and the similarity threshold for the seed-joining was set to be 0.60 for both rows and conditions. For the larger third dataset, we increase the number of partitions to $T = 25$. BCCA was run using the parameters suggested in the article, Bhattacharya and De (2009).

For all three datasets, we repeated the process of generating the data (keeping biclusters fixed), running the biclustering algorithms, and computing the recovery score five times. The scores reported in Table 2 show the average recovery scores from these five runs on the first dataset, where we vary the standard deviation of the background noise from 0 to 0.25. These results show that even in the highly idealized scenario of the first dataset, BCCA still makes errors, performing particularly poorly in the no-noise setting.

Table 2. Comparison of CLiP and BCCA on the First Simulation Dataset

	Recovery score					
Noise level (σ)	0	0.05	0.1	0.15	0.2	0.25
CLiP	1	1	1	1	1	1
BCCA	0.2	0.81	0.8	0.7	0.65	0.7

This can be attributed to the fact that without noise, the rows of the true biclusters are perfectly correlated on the entire set of conditions: by design, the entries for the complement conditions are all zero. BCCA does not eliminate these conditions in their search because it is only concerned with maximizing correlation, for which the addition of these zeroes has no effect.

For the second dataset, CLiP obtained an average recovery score of 0.85, and BCCA 0.20. In each of the five runs, CLiP recovered the true genes of at least one of the biclusters. In all but one case of inexact match, CLiP was off from the truth by one or two genes. By comparison, BCCA obtained only two true rows from each bicluster in all five simulations. Furthermore, the conditions found by CLiP also match those of the true biclusters in all but a few cases. In these cases, the results CLiP obtained are only off from the truth by at most two conditions.

On the third dataset, CLiP attained an average recovery score of 0.89. In comparison, BCCA attained an average recovery score of 0.03.

3.1.2. Biclusters with constant, coherent, and coherent evolution patterns. As described in the Introduction, many methods exist for identifying patterns which are versions of the linear pattern. We demonstrate the versatility of our method in this section by showing that CLiP performs well in capturing these other patterns. Specifically, we compare CLiP with the methods FABIA (Hochreiter et al., 2010), QUBIC (Li et al., 2009), ISA (Mclachian and Basford, 1998; Ihmels et al., 2004), and BCCA in recovering biclusters with constant, additive, and coherent evolution patterns. We note that for showing that the method can capture coherent patterns in general, it is enough to use datasets with additive patterns, since the multiplicative model can be expressed as an additive model after taking logarithms.

For datasets with the constant and additive patterns, we use the Prelic synthetic benchmarks (available at www.tik.ee.ethz.ch/sop/bimax). For a model of coherent evolutions, we use the QUBIC benchmark (available at csbl.bmb.uga.edu/$\sim$maqin/bicluster/benchmark.html).

The Prelic synthetic benchmarks consist of data for both the constant and additive patterns, under varying levels of noise and bicluster overlap. By assessing performance under different levels of noise, we get a sense of a method's robustness. Assessing performance on datasets with overlapping biclusters is a way to evaluate a method's ability to pick out sets of co-regulated genes in biological systems of varying complexity. The datasets with varying levels of noise are all 100×100 in size, and contain 10 non-overlapping biclusters of size 10×5. Noise is generated from a normal distribution with zero mean, and the different levels under consideration correspond to different standard deviations. For each level of noise, ten replicates are generated. Datasets with overlapping biclusters contain 10 square biclusters embedded in a 100×100 matrix with no noise. The biclusters vary in size from 10×10 (no overlap) to 18×18, where overlap occurs simultaneously on rows and conditions. For example, in the 15×15 case, any two biclusters have a 5×5 submatrix in common. The expression values for these biclusters are generated from an artificial model for gene regulation, the details of which can be found in Prelic et al. (2006).

The QUBIC benchmarks are similarly structured, consisting of 100×100 matrices embedded with two different coherent evolution patterns under varying levels of noise and overlap. For completeness, we also include comparison results for a constant pattern dataset from these benchmarks. As with the Prelic datasets, bicluster overlap occurs simultaneously on the rows and columns, and noise is generated from a zero-mean normal distribution under different standard deviation settings. Again, ten replicates are generated for each of these noise levels.

As before, we use the average module recovery score of Prelic et al. (2006) to assess performance, for which self-reported information from the QUBIC, FABIA, and ISA papers was available. Because BCCA was not evaluated on these datasets, we report results from running that method and CLiP. To produce the CLiP results that follow, we generate $T = 12$ random partitions $t = 3$ times and perform no seed filtering.

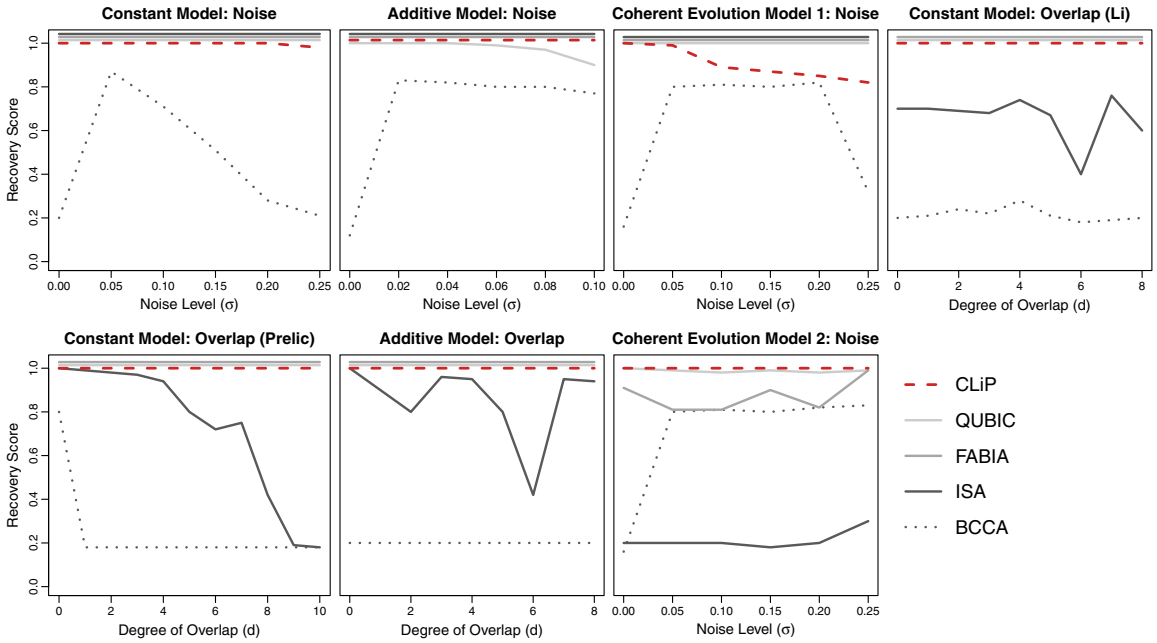

FIG. 2. Comparison of the recovery accuracy of five biclustering algorithms on simulation data for the constant, coherent, and coherent evolution models with varying levels of noise and overlap.

We also increase the number of generations in GA optimization to 500 from the default setting. Because the coherent evolution model deviates from a linear pattern, we set $\alpha = 0$ for datasets with that pattern.

Our findings from this method comparison are presented in Figure 2. They show that our method performs as well as existing methods on a wide range of expression patterns. The single exception is in one of the coherent evolution datasets from the QUBIC benchmarks, where CLiP performed slightly worse than QUBIC, FABIA, and ISA. In this dataset, the original bicluster pattern is comprised of genes possessing correlation of 1 or -1. However, the fitness function for our method is designed to capture consistent regulation, i.e., positive row correlations. In fact, as briefly mentioned in the Methods section, negative correlations are penalized. Thus, it is not surprising that CLiP did not perform as well on this dataset.

3.2. Real data

To assess our method's ability to identify biologically interesting sets of genes, we examine the common transcription factors in biclusters obtained on two well-studied yeast (*S. cerevisiae*) cell cycle datasets: the Tavazoie et al. (1999) dataset from before, with 2884 genes over 17 experiments, and expression data from Spellman et al. (1998), consisting of 6178 genes over 77 experiments.

As a performance measure, we calculate the average number of common transcription factors per bicluster, for simplicity taking only those in the promoter regions of genes. A high number of common transcription factors is an indicator that a set of genes is co-regulated. To identify these transcription factors, we used the program TOUCAN 2 to examine transcription factor binding sites in the proximal promoters of the genes from each bicluster. We consider a transcription factor ''common'' if it binds to the promoter regions of all of the genes in the bicluster.

To produce the results shown, a single iteration of CLiP was run with the number of partitions chosen so that each submatrix contained between 20–30% of all genes. No filtering was done after the GA optimization step. For BCCA, we followed the analysis outlined in Bhattacharya and De (2009), using the reported parameters.

The results from our comparison can be found in Supplementary Material. On the whole, CLiP identified more common transcription factors than BCCA. Furthermore, CLiP picked out transcription factors SCB and SW15, which BCCA missed. These are well-known transcription factors in DNA replication and the cell cycle. Because both datasets examine the yeast cell cycle, our results are more sensible.

In cross-checking our results with those reported, we also found some discrepancies between the BCCA results we obtained and those provided in their article. These findings are also reported in Supplementary Material.

Three additional datasets were used for validation in the article introducing BCCA. We could not report results for these due to technical issues in running TOUCAN 2 for these other datasets, stemming from the program's poor scalability. However, in keeping with the analyses performed in Bhattacharya and De (2009), we examined the number of functionally enriched categories per bicluster for all five datasets. A table summarizing these datasets and the details of this second analysis are presented in Supplementary Material.

3.3. Application to modENCODE time-course RNA-seq data

The modENCODE consortium generated RNA-Seq time-course data for developmental stages in *D. melanogaster* (common fruit fly) (The modENCODE Consortium et al., 2010) and *C. elegans* (a nematode worm) (Gerstein et al., 2010). We applied CLiP to the set of orthologous genes to identify any shared functional relationships in worm and fly. Because the functional roles of certain genes are more well-understood for some species than others, clustering with two species can aid in identifying gene function in the less studied case. In this application, CLiP identified a set of genes that may play a role in embryonic growth and oogenesis, suggesting maternal dependence. Worm maternal effect genes, however, are less well-characterized in the literature relative to fly. The results from our method thus provide possible leads for further investigation in worm.

The data is comprised of one RNA-Seq sample for each of the 30 developmental stages in fly and 14 developmental stages in worm; these developmental stages are illustrated in Figure 3. Read alignment for fly was performed with Bowtie (Langmead et al., 2009) and for worm, with MAQ (Li and Durbin, 2009). Using TreeFam (Li et al., 2006), 3574 pairs of one-to-one orthologous genes were identified between the species. For each orthologous gene pair, estimates of expression were obtained from Cufflinks (Trapnell, 2010), and the output fragments per KB per million (FPKMs) were normalized within both the fly and worm time courses. Using these normalized FPKMs, we constructed a 3574×44 matrix with orthologous genes pairs as rows, and developmental stages for fly and worm as columns.

The bicluster with largest fitness value found by running CLiP on this matrix is represented as a heatmap in Figure 4. This bicluster contains 178 orthologous gene pairs with 27 fly stages (i.e., all except ''Embryo 0–2h,'' ''Embryo 2–4h,'' and ''Embryo 4–6h'') and all 14 worm stages.

The fly time-course data shows that these genes are only highly expressed in the early embryonic and adult female stages. Conspicuously, these genes are lowly expressed in the adult male stage. These results are consistent with the characterization of maternal effect genes as critical in early embryo development and dependent on the genotype of the mother (Johnston, 2002; Jorgensen and Mango, 2002).

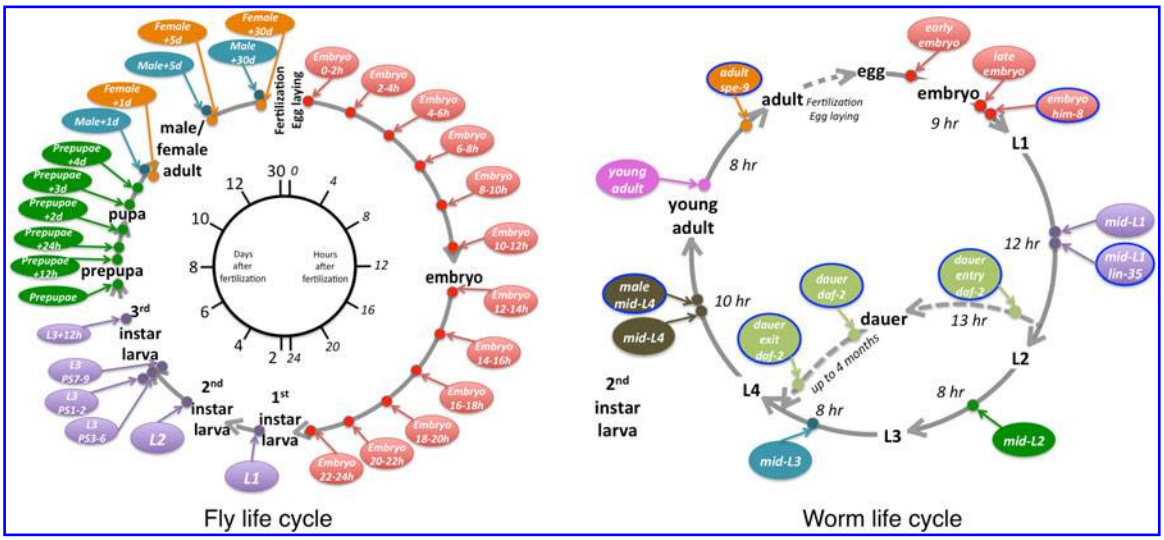

FIG. 3. The developmental stages of *D. melanogaster* and *C. elegans*.

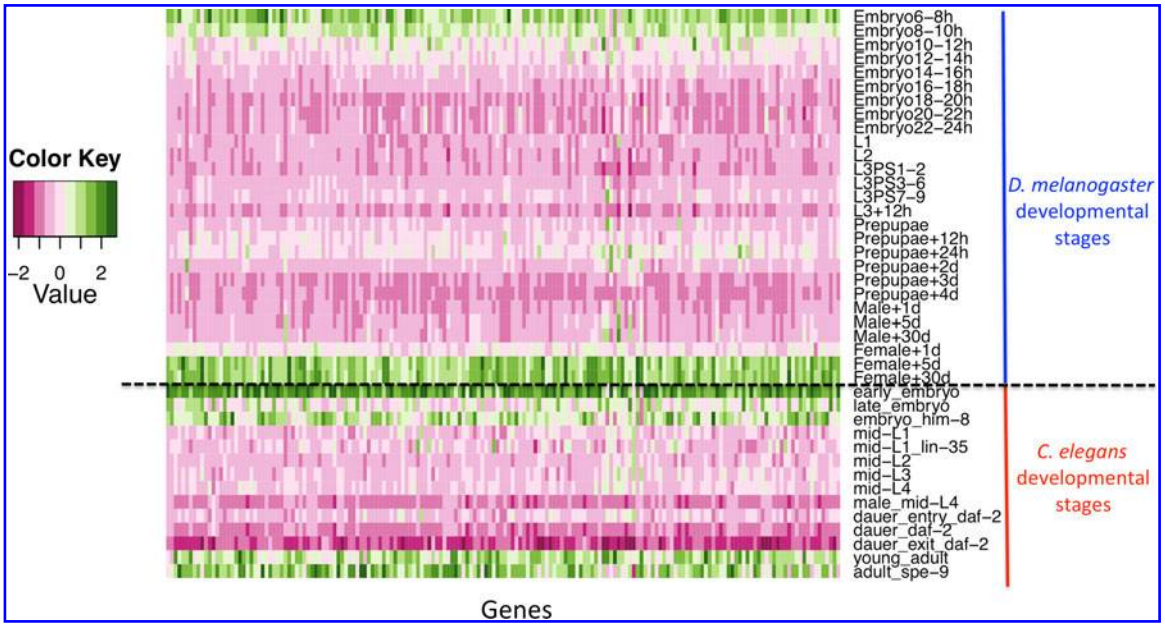

FIG. 4. The bicluster of best fitness obtained from running CLiP on modENCODE RNA-seq time-course data for orthologous genes in fly and worm. This bicluster contains 178 genes and 27 conditions.

We see analogous results for worm, a species that is dominantly hermaphrodite, i.e., they produce both eggs and sperm. As with fly, these genes are highly expressed in only the embryonic and adult stages; and, rather crucially, even under disrupted spe-9 function. Worms with this mutation experience defective sperm production (Singson et al., 1998), suggesting that the high expression of these genes can be attributed solely to oogenesis, or a maternal effect.

The similarity of the expression pattern between fly and worm suggests that these genes have similar functions in the two species.

To validate our findings, we performed Primary Gene Ontology analysis using the Database for Annotation, Visualization and Integrated Discovery (DAVID) software (Huang et al., 2009b,a). Through a gene set enrichment analysis, DAVID found that the genes in the identified bicluster are significantly enriched in two functional categories: ''oogenesis'' and ''embryonic development.'' We observed that 119 out of the 178 (66.9%) bicluster genes fell into these two categories, compared to 1649 out of the total 3574 (46.1%) orthologous genes were observed in these two categories. A Fisher's exact test shows that the found bicluster has a significantly higher percentage ($p \approx 10^{-8}$) of genes related to oogenesis and embryonic development compared to the entire set of orthologous genes considered. These findings corroborate our biological interpretation of the biclustering results. In addition, known maternal effect genes in fly were recovered in the results.

4. DISCUSSION

We have introduced a new method, CLiP, for finding biclusters with linear patterns. Real data application results show that CLiP performs well for both microarray and RNA-seq gene expression data.

An advantage of CLiP over existing techniques lies in the design of the fitness function. Rather than solely maximizing Pearson correlation, the fitness function rewards high contrast between a bicluster and its surrounding environment. This method eliminates the need for predeterminination of bicluster size. In addition, unlike BCCA, our method does not establish a heuristic correlation threshold during optimization. This facet of the BCCA optimization procedure could largely account for the poor performance we observed in our method comparison results.

Concerning computational efficiency, CLiP does not present a runtime advantage over existing methods. In fact, on the YCCD dataset of the Results section, we find that CLiP runs two times slower than BCCA.

This disadvantage does not diminish the contribution of our method, because CLiP offers a significant improvement over existing performance results. A detailed time complexity analysis and discussion of the runtime of our algorithm can be found in Supplementary Material.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for helpful comments. This work was supported by the National Institutes of Health (grant EY019094), the China National 973 Program (grant 2012CB821200), and the NSFC (grants 61134005, 60921001, 90916024, and 91116016).

DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

Aguilar-Ruiz, J. 2005. Shifting and scaling patterns from gene expression data. *Bioinformatics* 21, 3840–3845.

Banfield, J., and Raftery, A. 1993. Model-based Gaussian and non-Gaussian clustering. *Biometrics* 49, 803–821.

Ben-Dor, A., Chor, B., and Karp, R. 2003. Discovering local structure in gene expression data: the order-preserving submatrix problem. *J. Comput. Biol.* 10, 803–821.

Bhattacharya, A., and De, R. 2009. Bi-correlation clustering algorithm for determining a set of co-regulated genes. *Bioinformatics* 25, 2795–2801.

Busygin, S., Jacobsen, G., and Kramer, E. 2002. Double conjugated clustering applied to leukemia microarry data. Proc. *2nd SIAM ICDM Workshop Clustering High Dimensional Data*.

Califano, A., Stolovitzky, G., and Tu, Y. 2004. Analysis of gene expression microarrays for phenotype classification. *Proc. Int. Conf. Intel. Syst. Mol. Biol.* 194, 1625–1638.

Cheng, Y., and Church, G. 2000. Biclustering of expression data. *Proc. Int. Cof. Intell. Syst. Mol. Biol.* 8, 93–103.

Fraley, C., and Raftery, A. 2002. Model-based clustering, discriminant analysis and density estimation. *J. Am. Stat. Assoc.* 97, 611–631.

Gan, X., Liew, A., and Yan, H. 2005. Biclustering gene expression data based on a high dimensional geometric method. *Proc. Int. Conf. Mach. Learn. Cybernet.* 3388–3393.

Gar, X., Liew, A., and Yan, H. 2008. Discovering biclusters in gene expression data based on high-dimensional linear geometries. *BMC Bioinform.* 9, 209.

Gerstein, M.B., et al. 2010. Integrative analysis of the Caenorhabditis elegans genome by the modENCODE Project. *Science* 330, 1775–1787.

Getz, G., Levine, E., and Domany, E. 2000. Coupled two-way clustering analysis of gene microarray data. *Proc. Natl. Acad. Sci. USA* 97, 12079–12084.

Hartigan, J. 1972. Direct clustering of a data matrix. *J. Am. Stat. Assoc.* 67, 123–129.

Hartigan, J. 1975. *Clustering Algorithms*. John Wiley and Sons, Inc. New York.

Hochreiter, S., Bodenhofer, U., and Heusel, M. 2010. FABIA: factor analysis for bicluster acquisition. *Bioinformatics* 26, 1520–1527.

Huang, D., Sherman, B., and Lempicki, R. 2009a. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Res.* 37, 1–13.

Huang, D., Sherman, B., and Lempicki, R. 2009b: Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nat. Protoc.* 4, 44–57.

Ihmels, J., Friedlander, G., and Bergmann, S. 2002. Revealing modular organization in the yeast transcriptional network. *Nat. Genet.* 31, 370.

Ihmels, J., Bergmann, S., and Barkai, N. 2004. Defining transcriptional modules using large-scale gene expression data. *Bioinformatics* 20, 1993–2003.

Johnson, S. 1967. Hierarchical clustering schemes. *Psychometrika* 32, 241–254.

Johnston, D. S. 2002. The art and design of genetic screens: *Drosophila melanogaster. Nat. Rev. Genet.* 3, 176–188.

Jorgensen, E., and Mango, S. 2002. The art and design of genetic screens: *Caenorhabditis elegans. Nat. Rev. Genet.* 3, 356–369.

Klugar, Y., Basri, R., and Chang, J. 2003. Spectral biclustering or microarray data: coclustering genes and conditions. *Genome Res.* 13, 703–716.

Langmead, B., Trapnell, C., Pop, M., et al. 2009. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.* 10, R25.

Lazzeroni, L., and Owen, A. 2002. Plaid models for gene expression data. *Stat. Sin.* 12, 61–86.

Li, H., and Durbin, R. 2009. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* 25, 1754–1760.

Li, H., Lavin, M., and Master, R. 1986. Fast Hough transform: a hierarchical approach. *Comput. Vision Graphics Image Process.* 36, 139–161.

Li, H., Coghlan, A., and Ruan, J. 2006. TREEFAM: a curated database of phylogenetic trees of animal gene families. *Nucleic Acids Res.* 34, D572–D580.

Li, G., Ma, Q., and Tang, H. 2009. QUBIC: a qualitative biclustering algorithm for analyses of gene expression data. *Nucleic Acids Res.* 37, e101.

Liu, J., and Wang, W. 2003. Op-cluster: clustering by tendency in high dimensional space. *Proc IEEE Int. Conf. Data Mining* 187.

Madeira, S., and Oliveira, A. 2004. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 1, 24–45.

Mclachian, G., and Basford, K. 1998. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York.

Murali, T., and Kasif, S. 2003. Extracting conserved gene expression motifs from gene expression data. *Pac. Symp. Biocomput.* 8, 77–88.

Nepomuceno, J., Troncoso, A., and Aguilar-Ruiz, J. 2011. Biclustering of gene expression data by correlation-based scatter search. *BioData Mining* 4, 3.

Prelic, A., Bleuler, S., and Zimmermann, P. 2006. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics* 22, 1122–1129.

Segal, E., Battle, A., and Koller, D. 2003. Decomposing gene expression into cellular processes. *Pac. Symp. Biocomput.* 8, 89–100.

Sheng, Q., Moreau, Y., and Moor, B. 2003. Biclustering microarray data by Gibbs sampling. *Bioinformatics* 19, ii196–ii205.

Singson, A., Mercer, K., and L'Hernault, S. 1998. *The C. elegans* spe-9 gene encodes a sperm transmembrane protein that contains egf-like repeats and is required for fertilization. *Cell* 93, 71–79.

Spellman, P., Sherlock, G., Zhang, M., et al. 1998. Comprehensive identification of cell cycle–regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell* 9, 3273–3297.

Tamaryo, P., Slonim, D., and Mesirov, J. 1999. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci. USA* 96, 2907–2912.

Tanay, A., Sharan, R., and Shamir, R. 2002. Discovering statistically significant biclusters in gene expression data. *Bioinformatics* 19, S136–S144.

Tang, C., Zhang, L., Zhang, A., et al. 2001. Interrelated two-way clustering: an unsupervised approach for gene expression data analysis. *Proc. IEEE 2nd Int. Symp. Bioinform. Bioeng. Conf.* 41–48.

Tavazoie, S., et al. 1999. Systematic determination of genetic network architecture. *Nat. Gen.* 22, 281–285.

The modENCODE Consortium et al. Identification of functional elements and regulatory circuits by Drosophila modENCODE. 2010. *Science* 330, 1787–1797.

Trapnell, C. 2010. Transcript assembly and quantification by RNA-seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat. Biotech. Mol.*, 28, 511–515.

Wang, H., Wang, W., Yang, J., et al. 2002: Clustering by pattern similarity in large data sets. *Proc. ACM SIGMOD Int. Conf. Manage. Data* 394–405.

Yang, J., Wang, H., Wang, W., et al. 2003: Enhanced biclustering on expression data. *Proc. 3rd IEEE Int. Symp. Bioinform. BioEng.* 321–327.

Zhao, H., Liew, A., Xie, X., et al. 2008: A new geometric biclustering algorithm based on the Hough transform for analysis of large-scale microarray data. *J. Theoret. Biol.* 251, 264–274.

Address correspondence to:
Dr. Haiyan Huang
Department of Statistics
University of California
Berkeley, CA 94720

E-mail: hhuang@stat.berkeley.edu