Mining Shifting-and-Scaling Co-Regulation Patterns on Gene Expression Profiles

Xin Xu
National University of Singapore
xuxin@comp.nus.edu.sg

Ying Lu
University of Illinois, Urbana-Champaign
yinglu@uiuc.edu

Anthony K. H. Tung
National University of Singapore
atung@comp.nus.edu.sg

Wei Wang
U. of North Carolina, Chapel Hill
weiwang@cs.unc.edu

Abstract

In this paper, we propose a new model for coherent clustering of gene expression data called **reg-cluster**. *The proposed model allows (1) the expression profiles of genes in a cluster to follow any shifting-and-scaling patterns in subspace, where the scaling can be either positive or negative, and (2) the expression value changes across any two conditions of the cluster to be significant. No previous work measures up to the task that we have set: the density-based subspace clustering algorithms require genes to have similar expression levels to each other in subspace; the pattern-based biclustering algorithms only allow pure shifting or pure scaling patterns; and the tendency-based biclustering algorithms have no coherence guarantees. We also develop a novel pattern-based biclustering algorithm for identifying shifting-and-scaling co-regulation patterns, satisfying both coherence constraint and regulation constraint. Our experimental results show that the reg-cluster algorithm is able to detect a significant amount of clusters missed by previous models, and these clusters are potentially of high biological significance.*

1. Introduction

Advances in microarray technologies have made it possible to measure the expression profiles of thousands of genes in parallel under varying experimental conditions. Table 1 shows an example dataset. Each row of the table corresponds to a gene (denoted as g_i) while each column corresponds to a certain condition (denoted as c_j) in which gene expression level is measured. A subset of genes showing correlated co-expression patterns across a subset of conditions are expected to be functionally related. A natural task is to group such subsets of genes and conditions together.

One well-known characteristic of high-dimensional data is that data objects are not correlated in full dimensional space but correlated only in a subset of dimensions (subspace). The density-based *subspace clustering* algorithms [1, 2, 4, 15, 16, 21] assume data objects of the same cluster to be close in correlated subspace and assign each data object to only one cluster. Yet in gene expression data, a gene or a condition may be involved in multiple pathways. To allow overlap between gene clusters, pioneering biclustering algorithms such as [6] have been proposed. A later advancement, pattern-based biclustering algorithms [24, 25, 26] take into consideration the fact that genes with strong correlation do not have to be spatially close in correlated subspace.

In this paper, we focus on the more general shifting-and-scaling co-regulation patterns, which have received little attention so far.

1.1. Motivation

Existing pattern-based biclustering algorithms are only able to address pure shifting patterns or pure scaling patterns separately: as shown in Figure 1. After a single shifting or scaling, a pattern may coincide with another pattern. In Figure 1, the six patterns are of the relationships: $P1 = P2 - 5 = P3 - 15 = P4 = P5/1.5 = P6/3$. PCluster [24] and δ-cluster [25] assume that scaling patterns can be transformed to shifting patterns after a logarithm transformation on the whole dataset D, and focuses on shifting patterns only. Tricluster [26] focuses on scaling patterns only, assuming that after a global exponential transformation of D, shifting patterns will all be transformed into scaling patterns. Assume d_{ic} and d_{jc} are expression levels of gene g_i and g_j on condition c, s_1 and s_2 are the scaling and shifting factors respectively; their mathematical relationships are given as follows:

$$d_{ic} = s_1 * d_{jc} \Rightarrow logd_{ic} = logd_{jc} + logs_1 \text{ [24, 25]} \quad (1)$$

$$d_{ic} = d_{jc} + s_2 \Rightarrow e^{d_{ic}} = e^{d_{jc}} \cdot e^{s_2} \text{ [26].} \quad (2)$$

No existing pattern-based algorithms can handle dataset with shifting-and-scaling patterns of the form $d_{iC} = s_1 * d_{jC} + s_2$, by which the six cohesive patterns in Figure 1 can be grouped together simultaneously.

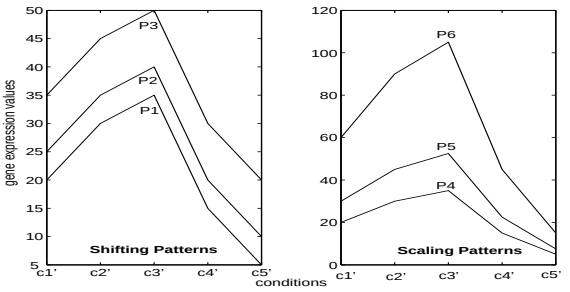


Figure 1. Previous Patterns

gene	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
g_1	10	-14.5	15	10.5	0	14.5	-15	0	-5	-5
g_2	20	15	15	43.5	30	44	45	43	35	20
g_3	6	-3.8	8	6.2	2	7.8	-4	2	0	0

Table 1. Running Dataset

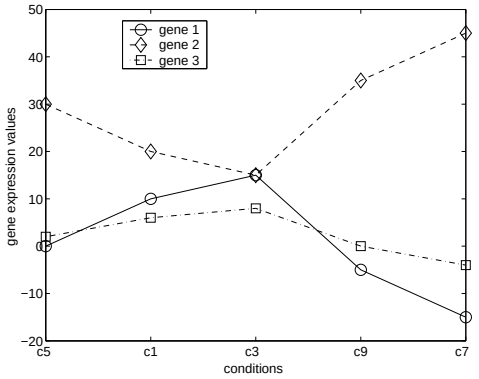


Figure 2. Our Shifting-and-Scaling Patterns

There are three problems the pattern-based algorithms and other existing biclustering algorithms have ignored:

• **Regulation Test**: Cheng and Church [6] state that the utmost important goal of gene expression data analysis is to find a set of genes showing strikingly similar *up-regulation* and *down-regulation* under a set of conditions, rather than simply to find a bicluster to perfectly cover the data. The pattern-based and tendency-based algorithms disregard the fact that patterns with smaller variations in expression values are probably of little biological meaning.

• **Pattern Universality**: Co-regulated genes may respond to environmental stimuli or conditions coherently, forming certain shifting-and-scaling patterns due to varying individual sensitivities. For instance, expression profiles of g_1 and g_3 of the running example (Table 1) in Figure 2 are shifting-and-scaling patterns: $d_{1,\{5,1,3,9,7\}} = 2.5 * d_{3,\{5,1,3,9,7\}} - 5$. Current pattern-based models [24, 25, 26] only validate a partial correlation, either a pure shifting pattern or a pure scaling pattern, special cases of the shifting-and-scaling pattern. Therefore, many co-regulation patterns would be missed by existing pattern-based algorithms, such as the one of real dataset in Figure 8.

• **Negative Correlation**: The complex biological system exhibits an even greater diversity in gene correlations than any existing subspace clustering and biclustering algorithms can capture. One is negative-correlation, i.e., when one gene has a high expression level, the expression level of the other gene is low and vice versa. Both positive-correlated genes and negative-correlated genes should be grouped together because they could be involved in the same biological pathway [9]. None existing subspace clustering algorithms has addressed the issue of negative correlation in a systematic way. Actually, negative correlation in subspace also pertains to the shifting-and-scaling pattern with a negative scaling factor, such as the relationship between g_2 and the other two genes in Figure 2, $d_{2,\{5,1,3,9,7\}} = -2.5 * d_{3,\{5,1,3,9,7\}} + 35 = -d_{1,\{5,1,3,9,7\}} + 30$.

1.2. Goal

To address the various problems that we have just discussed, we propose a new model called **reg-cluster**. The proposed model can better accommodate the regulation constraint and various correlation measures on gene expression profiles employed previously, including both positive and negative co-regulations. The proposed model also allows for shifting-and-scaling co-regulation as well as pure shifting and scaling one. Table 1 illustrates the expression levels of three genes under 10 conditions. As Figure 2 shows, g_1 and g_3 are strongly positively co-regulated, but g_2 strongly negatively co-regulated with g_1 and g_3 on conditions c_5, c_1, c_3, c_9 and c_7. The three genes form a candidate 3×5 reg-cluster before the regulation constraint is applied. A reg-cluster exhibits the following characteristics which are suitable for expression data analysis:

• increase or decrease of gene expression levels across any two conditions of a reg-cluster is significant with regard to the regulation threshold γ.

• increase or decrease of gene expression levels across any two conditions of a reg-cluster is in proportion, allowing small variations defined by the coherence threshold ϵ.

• genes of a reg-cluster can be either positively correlated or negatively correlated.

1.3. Challenges

In correlated subspace, positive-correlated genes and negative-correlated genes exhibit no spatial proximity at all. This makes it impractical to apply density-based subspace clustering algorithms [1, 2, 4, 15, 16, 21] and the mean-squared-residue-score based biclustering algorithm [6].

For pattern-based and tendency-based biclustering algorithms, there are three main challenges for reg-cluster discovery.

Naturally, the biggest challenge is the need of a novel coherent cluster model that can capture the more general shifting-and-scaling co-regulation patterns.

Another challenge is how to apply a non-negative regulation threshold. Tendency-based models of [3, 18, 19] are not suitable for adopting a regulation threshold γ. For example, the sequence model of [18], given the regulation threshold 0.8 for g_2 and the sorted expression levels $\{15, 20, 43, 43.5, 44\}$ on conditions c_2, c_{10}, c_8, c_4 and c_6, is unable to group non-regulated condition-pairs $c_8 - c_4$ and $c_4 - c_6$ together why not the regulated condition-pair $c_6 - c_8$.

The third challenge is negative co-regulation. Note that our scaling coefficient can be a negative real number. Existing pattern-based biclustering algorithms can not efficiently handle negative co-regulation problem, since the coexistence of positively and negatively correlated genes would lead to a rather large pScore [24] or expression ratio range [26].

The rest of paper is organized as follows. We review related work on gene expression data clustering in Section 2. Our reg-cluster model is presented in Section 3. We introduce our reg-cluster mining algorithm in Section 4. In Section 5, we present our experimental results. We conclude in Section 6.

2. Related Work

With the rapid advances of microarray technologies, large amounts of high-dimensional gene expression data are being generated, which poses significant computational challenges. Clustering is one of the most important tasks as similar expression profiles imply a related function and indicate the same cellular pathway [13]. Clustering algorithms may be classified into two big categories: full space clustering algorithms which evaluate the expression profile similarity of genes in all conditions, and subspace clustering algorithms which evaluate similarity in a subset of conditions.

The most commonly applied full space clustering algorithms on gene expression profiles are hierarchical clustering algorithms [10], self-organizing maps [12], and K-means clustering algorithms [23]. Hierarchical algorithms merge genes with the most similar expression profiles iteratively in a bottom-up manner. Self-organizing maps and K-means algorithms partition genes into user-specified k optimal clusters. Other full space clustering algorithms applied on gene expression data include Bayesian network [14] and neural network.

A common characteristic of high-dimensional data is that values of data objects may only be coherent under a subset of conditions. For this reason, density-based subspace clustering algorithms have been proposed [1, 2, 4, 15, 16, 21]. However, a common drawback of these density-based subspace clustering algorithms is that they assign each data object (gene) to one cluster only. Yet we know that a gene may participate in several biological pathways and thus it should be allowed to be assigned to multiple clusters. Biclustering algorithms provide an answer to this problem as it allows overlapping clusters. [6] is an example of a heuristic biclustering algorithm.

These algorithms require genes of the same cluster to be dense and close to each other when projecting into the correlated subspace. The more recent pattern-based and tendency-based biclustering algorithms, [3, 18, 19, 24, 25, 26] overcome the conventional constraint of spatial proximity and are able to identify shifting patterns, scaling patterns and synchronous-tendency patterns.

Current pattern-based and tendency-based algorithms focus on finding clusters with compatible expression profiles in subspace, disregarding regulation evaluation. The tendency-based biclustering algorithms like [18, 19] identify genes whose expression levels rise and fall synchronously in subspace without coherence guarantee. Existing pattern-based algorithms are limited to two simple types of patterns: pure shifting pattern and pure scaling pattern; they are unable to discover the more complicated shifting-and-scaling patterns. Another unaddressed issue is negative correlation, which is still confined to full space clustering at present [9, 17, 22].

Our reg-cluster model improves over previous pattern-based and tendency-based biclustering algorithms by addressing the more general shifting-and-scaling co-regulation patterns, allowing negative correlations as well as positive ones.

Our reg-cluster discovery algorithm gains efficiency by utilizing condition-wise enumeration strategy when searching representative regulation chains. Compared with previous condition-wise methods in gene expression analysis [20, 8, 7], reg-cluster per-

forms the condition-wise enumeration on ordered condition chains while the enumerated condition set of [20, 8, 7] is unordered.

3. The reg-cluster Model

3.1. Regulation Measurement

Suppose d_{ic_a} and d_{ic_b} are the expression levels of gene g_i under conditions c_a and c_b respectively. We could then say g_i is **up-regulated** from condition c_b to condition c_a, denoted as $\mathcal{R}eg(i, c_a, c_b) = Up$, if the increase in expression level exceeds its regulation threshold γ_i, as described in Equation 3. Alternatively, we say g_i is **down-regulated** from condition c_a to c_b, denoted as $\mathcal{R}eg(i, c_b, c_a) = Down$. We call c_b the **regulation predecessor** of c_a, denoted as $c_b \curvearrowleft c_a$, and c_a as the **regulation successor** of c_b for g_i, denoted as $c_a \curvearrowright c_b$.

$$\mathcal{R}eg(i, c_a, c_b) = \begin{cases} Up & \text{if } d_{ic_a} - d_{ic_b} > \gamma_i \\ Down & \text{if } d_{ic_a} - d_{ic_b} < -\gamma_i \end{cases} \quad (3)$$

In this paper, for ease of understanding, we assume the regulation threshold of g_i, γ_i, as a pre-defined percentage of the expression range of g_i in Equation 4, where n is the dimensionality of the expression dataset and γ is a user-defined parameter ranging from 0 to 1.0. We consider imposing a regulation threshold important for pattern validation, as it will help to distinguish useful patterns from noise. In practice, other regulation thresholds, such as the average difference between every pair of conditions whose values are closest [18], normalized threshold [17], average expression value [5], etc., can be used where appropriate.

$$\gamma_i = \gamma \times (MAX_{1 \le j \le n}(d_{ic_j}) - MIN_{1 \le j \le n}(d_{ic_j})), \quad (4)$$

The intuition behind using a local regulation threshold for different genes instead of a global one is that individual genes have different sensitivities to environmental stimulations. For instance, studies in [11] reveal that the magnitudes of the rise or fall in the expression levels of a group of genes inducible or repressible by hormone E2 can differ by several orders of magnitude. Current pattern-based and tendency-based models [3, 18, 19] can only cope with the extreme and probably biased case where $\gamma = 0$, and is constrained to the positive correlation. If $\gamma > 0$, these models become problematic, as we discussed in Section 1.

To support this general concept of regulation, instead of recording the regulation relationships between all possible pairs of C_n^2 conditions, we propose a new model, called $RWave^\gamma$ [1], which only keeps the regulation information of *bordering condition-pairs* for the genes in a wave-boosting manner with respect to γ. Figure 3 illustrates the $RWave^{0.15}$ model ($\gamma_1 = \gamma_2 = 4.5$ and $\gamma_3 = 1.8$) for the running example (Table 1). $c_5 - c_1$ is one bordering condition-pair for g_1, and that any condition c_i that lies on the left hand side of c_5 will guarantee to have a bigger difference than γ_1 when compared to any condition c_j that lies on the right hand side of c_1. The formal definition of the $RWave^\gamma$ model is given below.

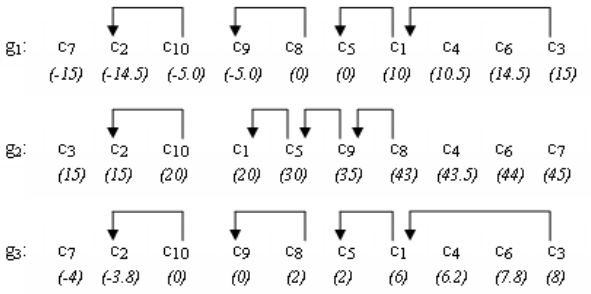


Figure 3. $RWave^{0.15}$ Models

Definition 3.1 $RWave^\gamma$
Given the regulation threshold γ, the $RWave^\gamma$ model of gene g_i on the set of conditions c_1, c_2, ..., and c_n is a non-descending ordering ($\preceq$) of the set according to their expression values with regulation pointers marking all the bordering regulation relationships such that for each regulation pointer pointing from c_b to c_a, we have, (1) $\forall c_p \succeq c_b$ and $\forall c_q \preceq c_a$, $\mathcal{R}eg(i, c_p, c_q) = Up$, denoted as $c_q \curvearrowleft c_p$; and (2) there is no other embedded pointer pointing from $c_{b'}$ to $c_{a'}$, such that $c_{a'} \succeq c_a$ and $c_{b'} \preceq c_b$, $\forall c_{p'} \succeq c_{b'}$ and $\forall c_{q'} \preceq c_{a'}$, we have $\mathcal{R}eg(i, c'_p, c'_q) = Up$, denoted as $c_{q'} \curvearrowleft c_{p'}$. □

Note that if $c_q \preceq c_p$ in g_i's $RWave^\gamma$ model, indicating $d_{iq} \le d_{ip}$, then c_q may not be c_p's regulation predecessor. Here, $\preceq$ and $\succeq$ indicate the ordering of the conditions while $\curvearrowleft$ and $\curvearrowright$ indicate the upward and downward regulation relationships of a condition-pair with respect to γ. Given the regulation threshold γ, the regulation relationship of any condition-pair of g_i can be easily inferred from its $RWave^\gamma$ model by simply checking whether there is a regulation pointer between the two conditions and what the pointer direction is. The conditions of a reg-cluster whose pairwise differences in expression levels are either upward or downward defined by γ must be separated by at least ONE regulation pointer in the $RWave^\gamma$ model of its genes, thus forming a "$\curvearrowleft$" or "$\curvearrowright$" linked regulation chain.

Besides, Lemma 3.1 ensures that $\forall c_k$ of a gene g_i, we can locate all the regulation predecessors and regulation successors of c_k for g_i efficiently by using the $RWave^\gamma$ model.

1 *RWave* stands for regulation wave

Lemma 3.1 Given the regulation threshold γ, a gene g_i and a condition c_a, let $c_p \curvearrowleft c_q$ be the nearest regulation pointer that is **before** c_a with respect to g_i. All conditions c_b such that $c_b \preceq c_p$ are all regulation predecessors of c_a with respect to g_i. Likewise, if $c_p \curvearrowleft c_q$ is the nearest regulation pointer that is **after** c_a, then all conditions c_b such that $c_q \preceq c_b$ are definitely the regulation successors of c_a for g_i.

Proof: Since the conditions are sorted in non-descending order of their expression levels, $c_b \preceq c_p \prec c_q \preceq c_a$ if $c_p \curvearrowleft c_q$ represents the nearest regulation pointer before c_a. Since the difference between the expression levels of c_p and c_q is greater than γ based on the definition of regulation pointer, we can also see that the difference between the expression levels of c_b and c_a is greater than γ. Thus c_b is considered to be the regulation predecessor of c_a. For the case in which $c_p \curvearrowleft c_q$ is the nearest regulation pointer after c_a, the same argument applies. □

Given the $RWave^{0.15}$ models in Figure 3, assume we want to find the regulation predecessors of c_6 for g_1, we simply follow the closest regulation pointer before it, which points from c_1 to c_5. c_7, c_2, c_{10}, c_9, c_8 and c_5 are exactly the regulation predecessors of c_6. We can also infer that there are no regulation successors of c_6 as no regulation pointer exists after c_6. Interested readers may refer to Table 1 for a more detailed analysis.

3.2. Coherence Measurement

Besides the regulation threshold γ, reg-cluster should be validated with the shifting-and-scaling coherency constraint ϵ. Assume d_{iY} and d_{jY} are two perfect shifting-and-scaling co-regulation patterns of g_i and g_j on condition set Y, then we there should exists s_1 and s_2 such that,

$$d_{iY} = s_1 * d_{jY} + s_2, \tag{5}$$

where s_1 and s_2 are the scaling and shifting factors respectively. The value of s_1 can be either positive ($s_1 > 0$), indicating d_{iY} and d_{jY} are **positively correlated** on Y, or negative ($s_1 < 0$), indicating d_{iY} and d_{jY} are **negatively correlated** on Y. Note that any subsequent shifting or scaling transformations on d_{iY} will not affect the general form given in Equation 5. Only the scaling and shifting factors may change values. As we can observe, the shifting patterns and scaling patterns addressed in [24, 25, 26] correspond to the two special cases of $d_{iY} = d_{jY} + s_2$ and $d_{iY} = s_1 * d_{jY}$ respectively.

Based on Equation 5, we can further infer the necessary and sufficient condition for the existence of shifting-and-scaling pattern.

Lemma 3.2 *Suppose d_{iY} and d_{jY} are the expression profiles of genes g_i and g_j on subspace Y, $Y = \{c_1, c_2, ..., c_n\}$, $d_{ic_1} < d_{ic_2} < ..... < d_{ic_n}$, and assume we choose c_1 and c_2 as the baseline condition-pair, then d_{iY} and d_{jY} are shifting-and-scaling patterns, either shifting-and-positive scaling or shifting-and-negative scaling, in subspace Y if and only if $\forall c_k$, $c_{(k+1)}$, $1 \leq k < n$,*

$$\frac{d_{ic_{k+1}} - d_{ic_k}}{d_{ic_2} - d_{ic_1}} = \frac{d_{jc_{k+1}} - d_{jc_k}}{d_{jc_2} - d_{jc_1}}. \tag{6}$$

Proof:

(1) If d_{iY} and d_{jY} are two shifting-and-scaling patterns, then $\exists s_1$ and s_2, $d_{iY} = s_1 * d_{jY} + s_2$. Furthermore, $\forall c_{(k+1)}$ and c_k, $1 \leq k < n$, we have $d_{ic_{k+1}} = s_1 * d_{jc_{k+1}} + s_2$ and $d_{ic_k} = s_1 * d_{jc_k} + s_2$, so $\frac{d_{ic_{k+1}} - d_{ic_k}}{d_{ic_2} - d_{ic_1}} = \frac{d_{jc_{k+1}} - d_{jc_k}}{d_{jc_2} - d_{jc_1}}$.

(2) On the other hand, if $\forall c_k$, $c_{(k+1)}$, $1 \leq k < n$ such that $\frac{d_{ic_{k+1}} - d_{ic_k}}{d_{ic_2} - d_{ic_1}} = \frac{d_{jc_{k+1}} - d_{jc_k}}{d_{jc_2} - d_{jc_1}}$, then $\forall c_p$, $c_q \in Y$, $p \neq q$, we have $\frac{d_{ic_p} - d_{ic_q}}{d_{ic_2} - d_{ic_1}} = \frac{(d_{ic_p} - d_{ic_{p-1}}) + (d_{ic_{p-1}} - d_{ic_{p-2}}) + ... + (d_{ic_{q+1}} - d_{ic_q})}{d_{ic_2} - d_{ic_1}} = \frac{(d_{jc_p} - d_{jc_{p-1}}) + (d_{jc_{p-1}} - d_{jc_{p-2}}) + ... + (d_{jc_{q+1}} - d_{jc_q})}{d_{jc_2} - d_{jc_1}} = \frac{d_{jc_p} - d_{jc_q}}{d_{jc_2} - d_{jc_1}}$. Therefore, $\frac{d_{ic_p} - d_{ic_q}}{d_{jc_p} - d_{jc_q}}$ is a constant for g_i and g_j, say s_1. Then $\forall c_p$, $c_q \in Y$, $p \neq q$, we have $d_{ic_p} = s_1 * d_{jc_p} - s_1 * d_{jc_q} + d_{ic_q}$, suggesting $d_{ic_p} - s_1 * d_{jc_p}$ being a constant as well, say s_2. So we can conclude that $d_{iY} = d_{jY} * s_1 + s_2$. □

Given Lemma 3.2, we need not check the coherence of reg-cluster on all combinations of pair-wise conditions, which was necessary in previous work. Instead, we simply check all adjacent condition-pairs c_k and c_{k+1} with regard to the baseline condition-pair, c_1 and c_2, according to a coherence threshold ϵ.

$$H(i, c_1, c_2, c_k, c_{k+1}) = \frac{d_{ic_{k+1}} - d_{ic_k}}{d_{ic_2} - d_{ic_1}}. \tag{7}$$

We can conclude that the expression profiles of the three genes in Figure 2 are shifting-and-scaling patterns on conditions c_7, c_9, c_5, c_1 and c_3 with each other because these three genes share exactly the same coherence scores: $\forall g_i \in \{g_1, g_2, g_3\}$, $H(i, c_7, c_9, c_7, c_9) = 1.0$, $H(i, c_7, c_9, c_9, c_5) = 0.5$, $H(i, c_7, c_9, c_5, c_1) = 1.0$ and $H(i, c_7, c_9, c_1, c_3) = 0.5$, with an order of either $c_7 \prec c_9 \prec c_5 \prec c_1 \prec c_3$ (g_1 and g_3) or $c_7 \succ c_9 \succ c_5 \succ c_1 \succ c_3$ (g_2).

We impose the coherence threshold ϵ to flexibly control the coherence of the clusters. In this way, we can ensure the variations in coherence scores, given in

Equation 7, are within ϵ for genes in the same cluster. Perfect shifting-and-scaling patterns correspond to the case where $\epsilon = 0$.

3.3. Model Definition and Comparison

By combining both the regulation constraint and the shifting-and-scaling coherence constraint, we now propose the definition of a reg-cluster.

Definition 3.2 *Reg-Cluster*
Given the regulation threshold γ and coherence threshold ϵ, a bicluster $C_{X\times Y}$, where X is a subset of genes and $Y = \{c_1, c_2, ..., c_n\}$ is the subset of correlated conditions such that $\forall g_i \in X$, either $d_{ic_1} < d_{ic_2} < ... < d_{ic_n}$ or $d_{ic_1} > d_{ic_2} > ... > d_{ic_n}$, is a reg-cluster if and only if:
(1) $\forall g_i \in X$, based on its $RWave^\gamma$ model, we have either $c_1 \curvearrowleft c_2 \curvearrowleft ..... \curvearrowleft c_n$, or $c_1 \curvearrowright c_2 \curvearrowright ..... \curvearrowright c_n$,
and (2) $\forall g_i, g_j \in X$, $\forall k$, $1 \leq k < n$, we have $|H(i, c_1, c_2, c_k, c_{k+1}) - H(j, c_1, c_2, c_k, c_{k+1})| \leq \epsilon$. □

In this way, with the reg-cluster model, we are able to identify all the significant shifting-and-scaling co-regulation patterns with regard to γ and ϵ. Two genes of a reg-cluster can be positively co-regulated if complying with the same regulation chain and negatively co-regulated if complying with inverted regulation chains.

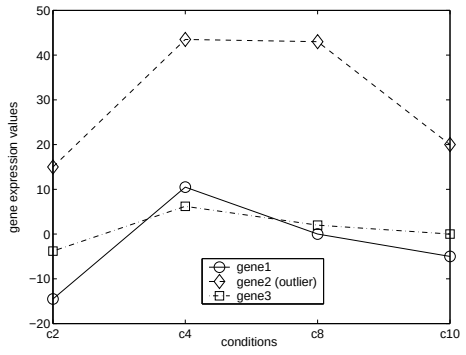


Figure 4. An Outlier

For a brief comparison between our reg-cluster model and previous models, consider the projection of the three genes in the running example on conditions c_2, c_4, c_8 and c_{10} in Figure 4, where $d_{3,\{2,4,8,10\}} = 0.4 * d_{1,\{2,4,8,10\}} + 2$ and there is no shifting-and-scaling relationship between g_2 and the other two genes. Given the regulation threshold $\gamma = 0.15$ and coherence threshold $\epsilon = 0.1$, our reg-cluster model can easily identify the outlier gene g_2 because (1) the $RWave^{0.15}$ model of g_2 indicates there are no regulation between c_4 and c_8; and (2) g_1 and g_3 have exactly the same coherence score along the four conditions while g_2 does not, i.e., $H(1, c_2, c_{10}, c_{10}, c_8) = H(3, c_2, c_{10}, c_{10}, c_8) = 0.5263$ but $H(2, c_2, c_{10}, c_{10}, c_8) = 4.6$, far beyond the allowed variation ϵ. In contrast, the pattern-based models discover no patterns, as there are no pure shifting or pure scaling relationships while the tendency-based models always cluster the three genes together because the three genes have exactly the same subsequence and tendency on the four conditions.

4. Algorithm

Input: $D = G \times C$: 2D dataset, $MinG$: minimum number of genes, $MinC$: minimum number of conditions, γ: regulation threshold and ϵ: coherence threshold.
Output: all validated reg-clusters w.r.t. γ, ϵ, $MinG$ and $MinC$: $\{C|C = X \times Y\}$ such that $C.X$ is the maximal gene set for the representative regulation chain $C.Y$.

* $RWave^\gamma$ model construction *\
for each gene $g_i \in G$ **do**
  sort the conditions $c_j \in C$ in non-descending order of d_{ij}.
  for each c_j in sorted order **do**
    find c_j's closest regulation predecessor c_k w.r.t. γ.
    if no regulation pointer exists between c_j and c_k **then**
      insert a new pointer $c_k \curvearrowleft c_j$ in g_i's $RWave^\gamma$ model.

* reg-cluster mining *\
$C.pX = C.nX = G$.
$C.Y = \emptyset$.
$C^2Set = \emptyset$.
$MineC^2(C, C^2Set)$.

Subroutine: $MineC^2(C, C^2Set)$.
Parameters:

$C.Y$: the current representative regulation chain;

$C.X$: the corresponding genes for $C.Y$;

C^2Set: the set of discovered validated reg-clusters.

Method:

1. **apply pruning (1): if** $|C.X| < MinG$, **then** return.
2. **apply pruning (3).(a): if** $|C.pX| < MinG/2$, **then** return.
3. assume $C.Y = c_{k1} \curvearrowleft c_{k2} ... \curvearrowleft c_{km}$, **if** $|C.Y| \geq MinC$ and $|C.X| \geq MinG$ and ($|C.pX| > |C.nX|$ or ($|C.pX| == |C.nX|$ and $k1 < k2$)) **then**
   apply pruning (3).(b):
   if C is already in C^2Set **then** return
   else output C to C^2Set.
4. Scan the $RWave^\gamma$ models of $C.pX$ when **applying pruning (2)** and store the condition candidates to $CandiSet$.
5. **for** each candidate condition $c_i \in CandiSet$ **do**
   find the subset of genes $X^{c_i} \subseteq C.X$ which match either $C.Y$+"$\curvearrowleft c_i$" or invert($C.Y$+"$\curvearrowleft c_i$") when **applying pruning (2)**;
   sort X^{c_i} on coherence score discrepancy $H(j, c_{k1}, c_{k2}, c_{km}, c_i)$ where $g_j \in X^{c_i}$;
   apply sliding window with minimum length $MinG$ and threshold ϵ on sorted X^{c_i};
   apply pruning (4):
   if no validated gene interval X'' **then** continue;
   for each validated X'' after sliding **do**
   $C'.Y = C.Y$+"$\curvearrowleft c_i$"; $C'.X = X''$;
   $MineC^2(C', C^2Set)$

Figure 5. reg-cluster Mining Algorithm

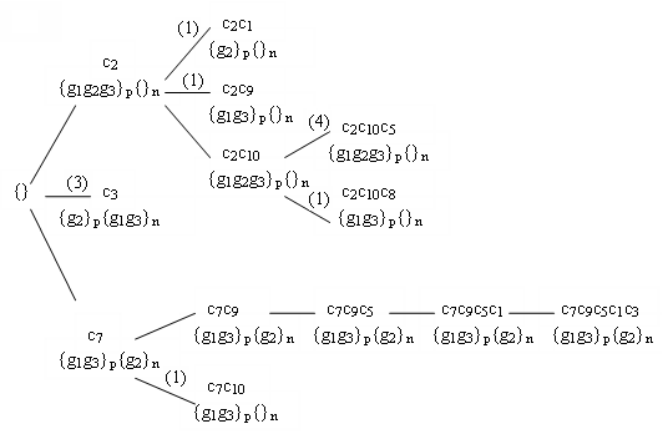


Figure 6. Enumeration Tree of Representative Regulation Chains w.r.t. $\gamma = 0.15$, $\epsilon = 0.1$, $MinG = 3$ and $MinC = 5$

The essential idea of our algorithm is to systematically identify the representative regulation chain for each validated reg-cluster. A **representative regulation chain** $C.Y = c_{k1} \curvearrowleft c_{k2} \curvearrowleft ... \curvearrowleft c_{km}$ (a series of conditions connected by regulation pointers) includes genes that are correlated or anti-correlated with the chain. We refer to them as the **p-members** $C.pX$ (gene complying with $C.Y$) and **n-members** $C.nX$ of the reg-cluster, respectively. We can conveniently obtain $C.pX$ by searching **along** the $RWave^{\gamma}$ model and $C.nX$ by searching in the **opposite** direction. Note that there are two regulation chains that a reg-cluster may satisfy: $C.Y$ and $invert(C.Y) = \{c_{k1} \curvearrowright c_{k2} \curvearrowright ... \curvearrowright c_{km}\}$.

To avoid redundancy and overlap of the output clusters, we assume that the *representative* regulation chain always captures the pattern of the majority of genes in a reg-cluster: the number of p-members is greater than or equal to the number of n-members. If the number of p-members is equal to that of the n-members, we assume the regulation chain starting with a predecessor of larger condition ID as the "representative". For instance, the representative regulation chain for the reg-cluster in Figure 2 is $c_7 \curvearrowleft c_9 \curvearrowleft c_5 \curvearrowleft c_1 \curvearrowleft c_3$ with its p-members $\{g_1, g_3\}$ and n-members $\{g_2\}$. The inverted $c_7 \curvearrowright c_9 \curvearrowright c_5 \curvearrowright c_1 \curvearrowright c_3$ is not a representative regulation chain.

In summary, our reg-cluster algorithm illustrated in Figure 5 performs a *bi-directional depth-first* search on the $RWave^{\gamma}$ models for representative regulation chains ($C.Y$) satisfying the user specified minimum number of genes $MinG$, minimum number of conditions $MinC$, regulation threshold γ, and coherence threshold ϵ. At any step, the candidate regulation successors for the partially enumerated representative regulation chain $C.Y$ are held in $CandiSet$. For each candidate $c_i \in CandiSet$, we locate the subset of genes $X^{c_i} \subseteq C.X$ which satisfy $C.Y \curvearrowleft c_i$ and sort them in non-descending order of the coherence score ($H(j, c_{k1}, c_{k2}, c_{km}, c_i)$, $g_j \in X^{c_i}$). Then we use a sliding window of the minimum length $MinG$ and coherence threshold ϵ to partition X^{c_i} into a set of validated maximal subset of genes X'', which may overlap. The same process $MineC^2()$ is applied to each partition C' ($C'.Y = C.Y \curvearrowleft c_i$ and $C'.X = X''$) recursively.

Figure 6 shows an example of representative regulation chain enumeration process. We apply the following pruning strategies:

(1) $MinG$ pruning: Whenever the total number of p-members and n-members of the current enumerated representative regulation chain is below $MinG$, we prune the search after this node, as further extension of the representative regulation chain will only reduce the number of genes.

(2) $MinC$ pruning: Whenever the estimated maximal length of the current enumerated representative regulation chain of a gene falls below $MinC$, we remove the gene from further consideration.

(3) Redundant pruning: (a) Whenever the number of p-members is below $MinG/2$ ($|C.pX| < MinG/2$), we prune the candidate reg-cluster because the number of p-members would be smaller than the number of n-members. (Any validated reg-cluster contains at least $MinG$ members.) (b) Whenever a validated reg-cluster is found to be repetitive (as a result of overlapping gene sets after applying the sliding window techniques), we prune the search because the search space rooted at this node is redundant.

(4) Coherence pruning: Whenever less than $MinG$ genes are coherent (defined by ϵ) at a node, we prune the search.

Note that with pruning strategies (2) and (3).(a), we only need to look at p-members of the current enumerated representative regulation chain $C.Y$ when searching for extending condition candidates.

Figure 6 is the representative regulation chain enumeration tree for the running example (Table 1) when $\gamma = 0.15$, $\epsilon = 0.1$, $MinG = 3$ and $MinC = 5$, which consists of six levels, 0, 1, ..., 5. The number on the tree edge indicates the pruning strategies applied. At the ith level, the bicluster subroutine tests all possible representative regulation chains of length i. The depth-first search starts from the root node initialized with an empty chain. At level 1, the only possible candidate conditions are c_2, c_3 and c_7. The rest conditions cannot grow any regulation chain of length 5 along the $RWave^{0.15}$ models (Figure 3). So we can prune the search on c_1, c_4, c_5, c_6, c_8, c_9 and c_{10} according to pruning strategies (2) and (3).(a). Moreover, we can prune the search following node c_3 using prun-

ing strategy (3).(a), because the number of p-members of the regulation chain c_3 is 1, which is smaller than $MinG/2$. Then, we grow the subtree of node c_2 with candidates c_1, c_9 and c_{10}, which are all possible conditions for extending a regulation chain of minimum length 5. With pruning strategy (1), we can prune the search after nodes c_2c_1 and c_2c_9. The only extensible child of node c_2 is c_2c_{10}, whose candidates are c_5 and c_8 with pruning strategy (3).(a). Node $c_2c_{10}c_5$ is pruned during coherence test with pruning strategy (4), since $H(1, c_2, c_{10}, c_{10}, c_5) = H(3, c_2, c_{10}, c_{10}, c_5) = 0.5263$ while $H(2, c_2, c_{10}, c_{10}, c_5) = 2$ and, therefore, no validated gene subset is discovered when sliding the window of minimum length 3 and $\epsilon = 0.1$. Node $c_2c_{10}c_8$ is pruned with pruning strategy (1). Again, we examine the p-members of node c_7 and find the candidates for further extension are c_9 and c_{10}. c_7c_{10} is pruned with strategy (1) and the only validated representative regulation chain discovered is $c_7 \curvearrowleft c_9 \curvearrowleft c_5 \curvearrowleft c_1 \curvearrowleft c_3$.

5. EXPERIMENTS

To evaluate the performance of our reg-cluster algorithm, we performed experiments on a series of synthetic datasets and real-life gene expression dataset on a 3.0-GHz Dell PC with 1G memory running Window XP.

As the running time of reg-cluster on real datasets are too short for in-depth analysis, we evaluate the efficiency of our algorithm on synthetic datasets, which are obtained with a data generator with three input parameters: number of genes ($\#g$), number of conditions ($\#cond$), and number of embedded clusters ($\#clus$). We set the default parameters of the data generator algorithm as $\#g = 3000$, $\#cond = 30$ and $\#clus = 30$. The synthetic dataset is initialized with random values ranging from 0 to 10. Then a number of $\#clus$ perfect shifting-and-scaling clusters of average dimensionality 6 and average number of genes (including both p-member genes and n-member genes) equal to $0.01 * \#g$ are embedded into the data, which are reg-clusters with parameter settings $\epsilon = 0$ and $\gamma = 0.15$.

We evaluate the effectiveness of our reg-cluster algorithm on a benchmark 2D yeast gene expression data [23], available at `http://arep.med.harvard.edu/biclustering/`. The 2D dataset contains the expression levels of 2884 genes under 17 conditions.

5.1. Efficiency

Given the default parameter setting of the data generator algorithm above, we test the scalability of reg-cluster by varying only one input parameter while keeping the other two as default. The average runtime of reg-cluster when we vary the parameters invoked with $MinG = 0.01 * \#g$, $MinC = 6$, $\gamma = 0.1$ and $\epsilon = 0.01$ is illustrated in Figure 7. As we can observe, the runtime of the reg-cluster algorithm is slightly more than linear in terms of the number of genes ($\#g$). It shows worse scalability with respect to the number of conditions ($\#cond$). This is because the reg-cluster algorithm may examine all possible permutations of conditions when looking for the representative regulation chains, but it only searches for the maximal sets of genes that are projected onto the enumerated (inverted) representative regulation chains. Typically, the number of conditions is much smaller than the number of genes. Figure 7 shows an approximately linear relationship between the runtime of the reg-cluster algorithm and the number of clusters ($\#cluster$).

5.2. Effectiveness

We ran the reg-cluster algorithm on the 2D 2884×17 yeast dataset with $MinG = 20$, $MinC = 6$, $\gamma = 0.05$ and $\epsilon = 1.0$; 21 bi-reg-clusters are output in 2.5 seconds, where the percentage of overlapping cells of a bi-reg-cluster with another one generally ranges from 0% to 85%. Note that we did not perform any splitting and merging of clusters. Due to space limit, we only report the details of three non-overlapping bi-reg-clusters with 21 genes and six conditions each.

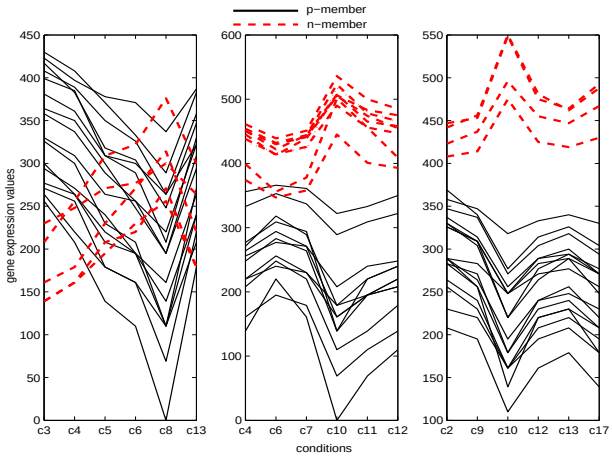
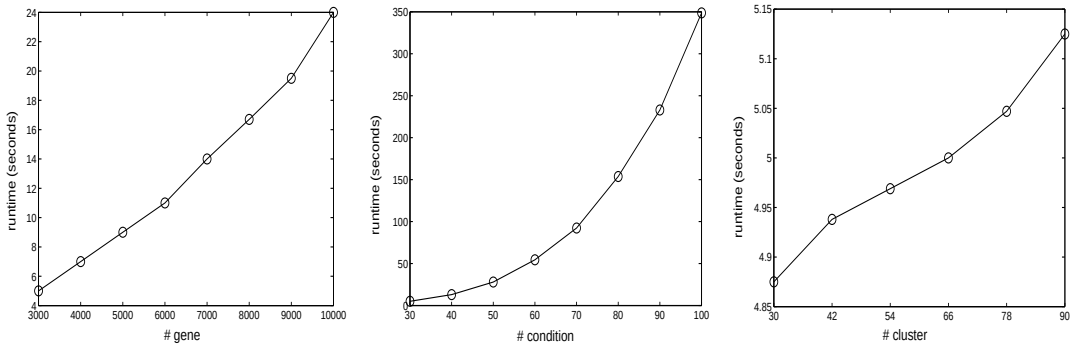


Figure 8. Three biclusters

Figure 8 illustrates the gene expression profiles for each of the three bi-reg-clusters. Our reg-cluster algorithm can successfully identify shifting-and-scaling patterns satisfying the regulation and coherence thresholds, where the scaling factor can be either positive or negative. For each bi-reg-cluster, we represent its p-members with black solid lines and its



(a) Varying $\#g$ (b) Varying $\#cond$ (c) Varying $\#clus$

Figure 7. Evaluation of Efficiency on Synthetic Datasets

Cluster	Process	Function	Cellular Component
c_1^2	DNA replication (p=3.64e-07)	DNA-directed DNA polymerase activity (p=0.01586)	replication fork (p=0.00019)
c_3^2	protein biosynthesis (p=0.00016)	structural constituent of ribosome (p=1.45e-07)	cytosolic ribosome (p=1.44e-08)
c_{13}^2	cytoplasm organization and biogenesis (p=5.72e-05)	helicase activity (p=0.00175)	ribonucleoprotein complex (p=0.0002)

Table 2. Top GO Terms of the Discovered Biclusters

n-members with red dashed lines. Obviously, the relationship between any two p-member genes or between any two n-member genes of the same cluster is shifting-and-positive-scaling while that between a p-member gene and a n-member gene is shifting-and-negative-scaling. As a remarkable characteristic of reg-clusters, crossovers can be observed frequently in the gene expression profiles of a pair of genes, resulting from the combination effects of shifting and scaling. In contrast, previous pattern-based biclustering algorithms [24, 25, 26] only allow pure shifting or pure positive-scaling patterns (but not a mixture of both) and hence fail to identify the three bi-reg-clusters.

We apply the yeast genome gene ontology term finder (`http://db.yeastgenome.org/cgi-bin/GO/goTermFinder`) on each discovered clusters to evaluate their biological significance in terms of associated biological processes, cellular components and gene function respectively. Table 2 shows the top GO terms of the three categories and the GO terms with the lowest p-values for the 3 bi-reg-clusters in Figure 8, which have been overlooked by previous work. Despite the relatively smaller number of genes with our regulation threshold $\gamma = 0.05$, the extremely low p-values suggest that the three bi-reg-clusters are of significant biological meaning in terms of biological process, cellular component and gene function.

Further experimental results show that our reg-cluster algorithm can identify a much broader range of biologically significant gene clusters. Each group of genes in these clusters show strikingly similar regulation under a subset of conditions.

6. Conclusion

In this work, we have overcome the problem of previous pattern-based biclustering algorithms which can only find either pure shifting or pure positive scaling patterns. We have proposed a general reg-cluster model for identifying arbitrary shifting-and-scaling co-regulation patterns, where the scaling can be either positive or negative. Unlike previous work, our algorithm also allows a flexible regulation threshold to quantify up or down regulation. The shifting-and-scaling patterns manifest a synchronous and proportional change of expression values in a subspace, and are able to capture both positive correlations and negative correlations among the genes in the subspace. We have developed a bi-directional depth-first algorithm which effectively and efficiently mine the reg-clusters using a novel $RWave^{\gamma}$ model. Our experimental results prove that our reg-cluster algorithm is able to discover a significantly number of biologically meaningful reg-clusters missed by previous work.

References

[1] C. C. Aggarwal and P. S. Yu. Finding generalized projected clusters in high dimensional spaces. In *Proc. of ACM SIGMOD Conf. Proceedings*, volume 29, 2000.

[2] C. C. Agrawal, C. Procopiuc, J. L. Wolf, P. S. Yu, and J. S. Park. Fast algorithms for projected clustering. In *Proc. of ACM SIGMOD Int. conf. on Management of Data*, 1999.

[3] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini. Discovering local structure in gene expression data: the ordering-preserving submatrix problem. In *Recomb*, 2002.

[4] C. Bohm, K. Kailing, P. Kroger, and A. Zimek. Computing clusters of correlation connected objects. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, 2003.

[5] T. Chen, V. Filkov, and S. S. Skiena. Identifying gene regulatory networks from experimental data. In *Recomb*, 1999.

[6] Y. Cheng and G. M. Church. Biclustering of expression data. In *Procs of the Eighth International Conference on Intelligent Systems for Molecular Biology*, 2000.

[7] G. Cong, K.-L. Tan, A. K. H. Tung, and X. Xu. Mining top-k covering rule groups for gene expression data. *In the 24th ACM SIGMOD International Conference on Management of Data*, 2005.

[8] G. Cong, A. K. H. Tung, X. Xu, F. Pan, and J. Yang. Farmer: Finding interesting rule groups in microarray datasets. *In the 23rd ACM SIGMOD International Conference on Management of Data*, 2004.

[9] I. S. Dhillon, E. M. Marcotte, and U. Roshan. Diametrical clustering for identifying anti-correlated gene clusters. *Bioinformatics*, 19, 2003.

[10] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analisis and display of genome-wide expression patterns. In *Proc. Natl. Acad. Sci. USA*, volume 95, pages 14863–14868, 1998.

[11] K. R. C. et al. Global analysis of ligand sensitivity of estrogen inducible and suppressible genes in mcf7/bus breast cancer cells by dna microarray. In *Proceedings of the National Academy of Sciences of the United States of America*, 2003.

[12] P. T. et al. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. In *Proc. Natl. Acad. Sci. USA*, volume 96, pages 2907–2912, 1999.

[13] T. R. H. et al. Functional discovery via a compendium of expression profiles. *Cell*, 102, 2000.

[14] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using bayesian network to analyze expression data. In *Recomb*, pages 127–135, 2000.

[15] A. Hinneburg and D. A. Keim. Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In *Proc. 1999 Int. Conf. Very Large Data Bases*, 1999.

[16] C. hung Cheng, A. W. chee Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining*, 1996.

[17] L. Ji and K.-L. Tan. Mining gene expression data for positive and negative co-regulated gene clusters. *Bioinformatics*, 20, 2004.

[18] J. Liu and W. Wang. Op-cluster: Clustering by tendency in high dimensional space. In *Proc. of the Third IEEE International Conference on Data Mining*, 2003.

[19] J. Liu, W. Wang, and J. Yang. Gene ontology friendly biclustering of expression profiles. In *Computational Systems Bioinformatics*, 2004.

[20] F. Pan, G. Cong, A. K. H. Tung, J. Yang, and M. J. Zaki. Carpenter: Finding closed patterns in long biological datasets. In *Proc. 2003 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[21] C. M. Procopiuc, M. Jones, P. K. Agarwal, and T. M. Murali. A monte carlo algorithm for fast projective clustering. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, 2002.

[22] J. Qian, M. Dolled-Filhart, J. Lin, H. Yu, and M. Gerstein. Beyond synexpression relationships: Local clustering of time-shifted and inverted gene expression profiles indentifies new, biologically relevant interactions. In *Journal of Molecular Biology*, 2001.

[23] S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church. Systematic determination of genetic network architecture. In *Nature Genetics*, volume 22, pages 281–285, 1999.

[24] H. Wang, W. Wang, J. Yang, and P. S. Yu. Clustering by pattern similarity in large data sets. In *Proc. of the 2002 ACM SIGMOD international conference on Management of data*, 2002.

[25] J. Yang, W. Wang, H. Wang, and P. Yu. δ-clusters: Capturing subspace correlation in a large data set. In *Proc. of the 18th International Conference on Data Engineering*, 2002.

[26] L. Zhao and M. J. Zaki. Tricluster: An effective algorithm for mining coherent clusters in 3d microarray data. In *Proc. of the 2005 ACM SIGMOD international conference on Management of data*, 2005.