this is the extended version of the paper
accepted by ICDE2009

Exploiting Domain Knowledge to Improve Biological Significance of Biclusters with Key Missing Genes

Jin Chen #1, Liping Ji #2, Wynne Hsu *3, Kian-Lee Tan *4, Seung Y. Rhee #5

#*Department of Plant Biology, Carnegie Institution for Science*
260 Panama Street, Stanford, CA 94305
**Department of Computer Science, National University of Singapore*
Law Link, Singapore 117590

1,2{chenjin, jiliping}@stanford.edu 3,4{whsu,tankl}@comp.nus.edu.sg 5rhee@acoma.stanford.edu

***Abstract*— In an era of increasingly complex biological datasets, one of the key steps in gene functional analysis comes from clustering genes based on co-expression. Biclustering algorithms can identify gene clusters with local co-expressed patterns, which are more likely to define genes functioning together than global clustering methods. However, these algorithms are not effective in uncovering gene regulatory networks because the mined biclusters lack genes that may be critical in the function but may not be co-expressed with the clustered genes. In this paper, we introduce a biclustering method called *SKeleton Biclustering* (SKB), which builds high quality biclusters from microarray data, creates relationships among the biclustered genes based on Gene Ontology annotations, and identifies genes that are missing in the biclusters. SKB thus defines inter-bicluster and intra-bicluster functional relationships. The delineation of functional relationships and incorporation of such missing genes may help biologists to discover biological processes that are important in a given study and provides clues for how the processes may be functioning together. Experimental results show that, with SKB, the biological significance of the biclusters is considerably improved.**

I. Introduction

Gene expression clustering is one of the key steps in gene functional analysis. Genes that have similar patterns of expression can be clustered together, and are considered to be functionally related [1]. The gene clusters can thus help formulate new hypotheses from high-throughput experimental data.

Conventional clustering algorithms [2] cluster genes based on all conditions tested whereby the feature space is globally shared by all resulting clusters. However, in many cellular processes, many genes are usually co-expressed only under certain experimental conditions, but behave almost independently under other conditions [3]. Hence, discovering local co-expressed patterns becomes the key in uncovering genetic pathways that are not apparent when clustered globally. Therefore, biclustering algorithms [4–8] have been proposed to capture a subset of genes that may function together under a specific condition by simultaneously clustering both the genes and experimental conditions together. As highlighted in [5], bicluster identification is essential in revealing gene regulatory networks.

While existing biclustering algorithms can detect biclusters with local co-expressed patterns, they are not effective in uncovering gene regulatory networks or genetic pathways, mainly due to the following two reasons.

First, it is the relationship among clusters and the relationship among the genes within a cluster rather than the sets of clusters that contribute towards better interpretation of the overall picture of genetic pathways and gene regulatory networks [9]. Although hierarchical biclustering algorithms [8] can reveal the inter-bicluster (relationships among biclusters) relationships, existing biclustering algorithms [4–8], to our knowledge, cannot identify the intra-bicluster (relationships among genes within a single bicluster) relationships. Methods to investigate how the genes are functionally associated within a bicluster and to improve biclustering performance by reinforcing the gene functional associations have not yet been developed.

Second, biclustering methods solely based on microarray data would inevitably miss certain functionally related genes, no matter how well the algorithms are tuned. This is because: 1) not all of the functionally related genes necessarily co-express significantly; 2) unavoidable experimental noises or missing values may occur in microarray data. Hence, certain genes cannot be grouped into a bicluster, although they are functionally similar to the biclustered genes. Without these missing genes, the ability to associate functions among the biclustered genes would be substantially reduced, resulting in weak overall intra-bicluster relationships. For example, transcription factors (TF) are usually not co-expressed substantially with their target genes, and the functional associations among their target genes may not be made if a key TF is missing from the bicluster. The gene pathways or regulatory networks built upon such biclusters would then be incomplete and disconnected. Therefore, a systematic analysis on each bicluster to identify new genes as false negatives would help to better interpret the intra-bicluster relationships, hence improve the biological significance of the biclusters. However, to the best of our knowledge, no algorithm to date exists, which can identify such missing genes and missing gene associations.

In summary, a biclustering algorithm that could i) build high quality biclusters from microarray data, ii) reveal inter-

and intra-bicluster relationships, and iii) identify missing genes and missing gene associations, would be valuable in interpreting and forming hypotheses with microarray data.

In this paper, we propose a novel biclustering algorithm called *SKeleton Biclustering* (SKB) to mine biclusters. SKB not only builds biclusters and reveals bicluster skeleton (inter-bicluster and intra-bicluster relationships), but also identifies relevant missing genes that can bridge the functionally distinct biclustered genes, in order to uncover the otherwise hidden gene associations within biclusters. Figure 1 shows the framework of SKB. Overall, SKB has three phases. In phase 1, a hierarchical biclustering method is introduced to generate biclusters from microarray data. The inter-bicluster relationships are revealed by a hierarchical tree. In phase 2, each bicluster is converted into either one connected graph or a set of separated subgraphs by linking the genes that have similar biological features. In the graph, the distance between any two genes is measured based on biological domain knowledge, including Gene Ontology (GO) annotation [10], cis-elements and others. Connections of the genes within a bicluster based on their functional similarity define the initial intra-bicluster relationships. In phase 3, a graph mining method is proposed to efficiently add new genes, if any, to each bicluster, to reconnect individual subgraphs generated in phase 2, in order to enhance the intra-bicluster relationship. Therefore, the overall functional connectivity of the bicluster is increased.

Comparing with previous methods, we have made contributions in the following four aspects.

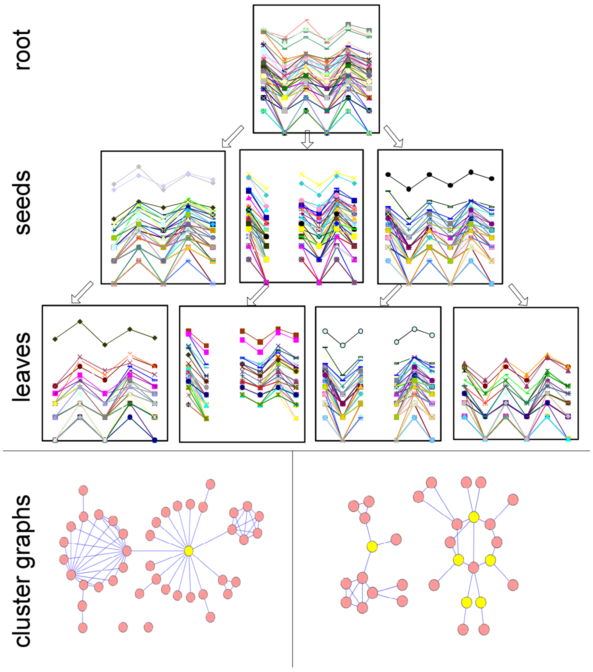


Fig. 1. Framework of SKB. In the upper part, SKB mines biclusters from microarray data and reveals the inter-bicluster relationship with a hierarchical tree. In the tree, each node is a bicluster, shown with gene expression changing trend. In the lower part, SKB reveals the intra-bicluster relationships by identifying missing genes (yellow colored) that could bridge the functionally distinct genes in each bicluster, so as to uncover the otherwise hidden gene associations.

1. New Concept of Biclustering. We are the first to propose the concept to mine the skeleton (*i.e.*, inter-cluster and intra-cluster relationships) of biclusters, rather than to mine the gene biclusters only. We are also the first to propose a new way to improve the bicluster quality by adding new objects into the biclusters to reinforce such intra-bicluster relationships.

2. New Methodology to Employ Domain Knowledge. It is generally difficult to integrate mining data with domain knowledge in an unsupervised learning model. Work to date either combines domain knowledge with mining data to form a complicated feature space, or takes domain knowledge as a post-processing filter. However, the former approach suffers from the complicated combination of heterogeneous features, while the latter approach results in data loss. In this paper, we employ the domain knowledge as an independent data space, and map the mining results from mining data to such a space to further improve the mining performance.

3. Improved Biological Significance. We are the first to improve the biological significance of biclusters by reinforcing the gene functional associations. In biological processes, co-expressed genes with distinct GO annotations may not “directly interact” with each other in gene pathways. For these genes, certain key genes, which are missing from biclusters due to their different expression patterns, can functionally connect the genes in the clusters together to form a unified functional module. As such, existing biclustering algorithms focusing on co-expression inevitably miss these key genes, resulting in weakened gene functional association in the biological networks. Our method, however, can easily find such functionally related genes that are not part of the biclusters and reinforce the gene function associations.

4. Efficient Algorithm. Both gene biclustering and missing gene identification are computationally expensive. We introduce an appropriate framework to achieve the goal effectively.

Experimental results on Yeast cell cycle [4] and Arabidopsis cold-response microarray datasets [11] show that SKB can build a clear structure of the inter-bicluster and intra-bicluster relationships. Consequently, the mined biclusters have significantly higher biological meaning than existing methods.

The rest of the paper is organized as follows. In the next section, we identify the challenges by providing a brief survey of biclustering and bicluster analysis methods with GO annotations. In Sections III, IV and V, we introduce the three phases of SKB. The algorithm is tested on Yeast and Arabidopsis microarray data in Section VI. Finally, we draw the conclusion and propose future direction in Section VII.

II. Related work

The “biclustering” algorithm on microarray data was first introduced by [4] to simultaneously cluster both genes and experimental conditions, which captures the coherence of a subset of gene under a subset of experimental conditions. In [4], the biclustering algorithm begins with the original microarray matrix and iteratively masks out null values and biclusters that have been discovered. The node-deletion and node-addition algorithms are introduced to find sub-matrices in expression data that have low mean squared residue (*MSR*) score. Let $X \subseteq I$ and $Y \subseteq J$ be subsets of genes and conditions. The *MSR* of submatrix $A_{X,Y}$ is defined as $H(X,Y) = \frac{1}{|X||Y|} \sum_{x \in X, y \in Y} (d_{xy} - d_{xY} - d_{Xy} + d_{XY})^2$ where $d_{xY} = \frac{1}{|Y|} \sum_{y \in Y} d_{xy}$, $d_{Xy} = \frac{1}{|X|} \sum_{x \in X} d_{xy}$, $d_{XY} = \frac{1}{|X||Y|}$ are the row and column means and the means in the submatrix A_{XY}. A submatrix A_{XY} is called a δ-bicluster if $H(X,Y) \leq \delta$ for some $\delta > 0$.

δ-cluster model [6] was proposed to further accelerate the biclustering process. The δ-cluster model incorporates null values and proposes a move-based algorithm (FLOC). FLOC starts with choosing initial biclusters called "seeds" randomly from the original matrix and proceeds with iterative gene/condition deletion and addition, aiming at achieving the best potential mean squared residue score reduction.

Another work accelerating the biclustering process by proposing a depth-first algorithm to mine "pClusters" [5]. This method clusters the dataset row-wise as well as column-wise to find pClusters that satisfy a user specified minimum *pScore*. Given $x, y \in X$, and $a, b \in Y$, the *pScore* of a 2×2 matrix is defined as:
$$pScore\left(\begin{bmatrix} d_{xa} & d_{xb} \\ d_{ya} & d_{yb} \end{bmatrix}\right) = |(d_{xa} - d_{xb}) - (d_{ya} - d_{yb})|$$

Pair (X,Y) forms a δ-pCluster if for any 2×2 submatrix A in (X,Y), $pScore(A) \leq \delta$ for some $\delta > 0$.

More recently, a deterministic algorithm DBF was proposed [7] to further improve the quality and efficiency of biclustering. DBF employs a frequent closed pattern mining algorithm to generate "good seeds" with better pattern similarity and then refines the seeds by adding genes and conditions to achieve a low *MSR* score as well as large bicluster volume. A minimum row variance threshold is set to remove biclusters with trivial changes in trends.

While these algorithms can generate biclusters with similar trends, they are limited in several ways. First, these schemes typically employ a similarity score (*e.g.*, *MSR* and *pScore*) to determine the quality of biclusters. However, the similarity scores cannot adequately capture the trend consistency of biclusters. Second, these algorithms usually generate biclusters based on selected "seeds" that cover only a small part of the whole dataset. As such, interesting patterns may be missed resulting in loss of relevant information. Third, the seed improvement process follows the hill-climbing paradigm and can involve significant amount of computation.

To overcome all of these limitations, the quick hierarchical biclustering algorithm (QHB) [8] was recently proposed to efficiently mine biclusters with both consistent trends and trends with similar degrees of fluctuations. It also produces a hierarchical tree to reveal the inter-bicluster relationships (details in Section III).

However, no biclustering algorithm to date exists that can identify the intra-bicluster relationships. Moreover, as stated earlier, the biclustering methods solely based on microarray data would inevitably miss certain genes that are functionally related but not by expression pattern and thus weaken the overall gene relationships. Hence, incorporating biological domain knowledge in clustering and cluster analysis [12–14] has been increasingly recognized as a reliable way to enhance the interpretability of biclustered genes. Gene Ontology(GO) annotation [10] has often been integrated as such domain knowledge for its high biological accuracy and interpretability.

The Gene Ontology project is a collaborative effort to construct and use ontologies to facilitate the systematic annotation of genes and their products in a wide variety of organisms [10]. The gene ontologies have now been accepted as the *de facto* language for the description of attributes of biological entities in three key domains that are shared by all organisms, namely molecular function, biological process and cellular component. In each of these domains, the corresponding GO ontology is structured as a directed acyclic graph to reflect the complex hierarchy of biological terminologies. Mathematically, suppose T is a set of GO terms, we say term t_i is a direct child of term t_j, if and only if t_i is a type ("is-a" relationship) or a component ("part-of" relationship) of t_j $(t_i, t_j \in T)$. GO annotations are usually used in clustering and cluster analysis in the following three ways [12–14].

First, GO annotation is often used as additional features in gene distance measures [12] to improve bicluster performance by incorporating genes that are functionally related but not co-expressed. However, this approach suffers from complicated combination of heterogenous features. Furthermore, owing to incomplete biological knowledge, such a combined metric is biased towards known information [15].

Second, GO annotation has been incorporated as an enrichment measure after processing the gene expression data [13]. A bicluster is functionally enriched for an attribute if the proportion of the biclustered genes known to have that attribute exceeds the number that could reasonably be expected from random chance in the reference dataset. The GO term enrichment measure can be used to summarize the biological knowledge of a cluster to filter out unenriched biclusters. But it cannot improve the performance of the enriched biclusters.

Third, GO could be used to analyze biclustering results in a graphical way by providing scoring functions and an easily-interpretable image to show the relationship within each cluster [14]. With the graph, one can analyze bicluster structures and evaluate biclusters from the topology aspect of the graphs. The graphical method has the potential to facilitate the interpretation of the bicluster structure and differentiate the biclustered genes from the topological view. However, the resulting graph is either one connected graph with many weak links or many small subgraphs if only strong links are allowed. This is because the biclustered genes are usually highly diverse in functionality and certain function-related key genes may be excluded from the biclusters.

To better use GO annotation to increase biclustering performance, we propose a biclustering algorithm called *Skeleton Biclustering* (SKB) that contains three phases: hierarchical biclustering, cluster graph generation and new gene identification. These will be introduced in the following three sections.

III. SKB Phase 1, Biclustering and Inter-Cluster Relationship Identification

We employ an efficient top-down hierarchical biclustering algorithm QHB [8] as the first phase of SKB to build high

TABLE I
ORIGINAL DATA MATRIX A

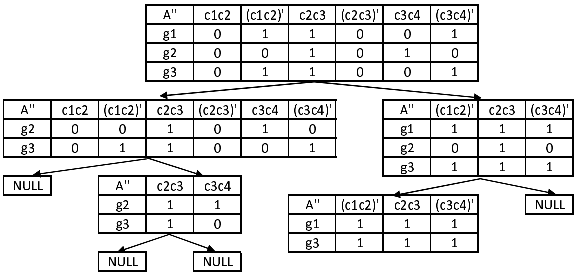
A	c_1	c_2	$c3$	$c4$
g_1	2.4	2.95	2.45	2.99
g_2	1.95	1.71	1	0.29
g_3	0.5	1.1	0.38	1.56

TABLE II
SLOPE ANGLE MATRIX A'

A'	c_1c_2	c_2c_3	c_3c_4
g_1	$28.81°$	$-26.57°$	$28.37°$
g_2	$-13.50°$	$-35.37°$	$-35.37°$
g_3	$30.96°$	$-35.75°$	$49.72°$

TABLE III
BINARY MATRIX A'' $(threshold = 26.5°)$

A''	c_1c_2	$(c_1c_2)'$	c_2c_3	$(c_2c_3)'$	c_3c_4	$(c_3c_4)'$
g_1	0	1	1	0	0	1
g_2	0	0	1	0	1	0
g_3	0	1	1	0	0	1



A''	c1c2	(c1c2)'	c2c3	(c2c3)'	c3c4	(c3c4)'
g1	0	1	1	0	0	1
g2	0	0	1	0	1	0
g3	0	1	1	0	0	1

A''	c1c2	(c1c2)'	c2c3	(c2c3)'	c3c4	(c3c4)'
g2	0	0	1	0	1	0
g3	0	1	1	0	0	1

A''	(c1c2)'	c2c3	(c3c4)'
g1	1	1	1
g2	0	1	0
g3	1	1	1

A''	c2c3	c3c4
g2	1	1
g3	1	0

A''	(c1c2)'	c2c3	(c3c4)'
g1	1	1	1
g3	1	1	1

Fig. 2. Partitioning process of matrix A'' in Table III in the running example.

quality biclusters from microarray data. QHB delivers biclusters with consistent trends and produces a hierarchical tree to reveal the inter-bicluster relationships. For completeness, we briefly describe QHB in this section with a running example.

A. QHB Framework

First, the microarray matrix A in Table I, where rows represent genes and columns represent experimental conditions, is transformed into a slope angle matrix A' in Table II to determine the fluctuating degrees of trends when the experimental condition changes. Then A' is transformed into a binary matrix A'' in Table III where the rising(01) and falling(10) trends are separated into two consecutive columns, while trends with trivial change(00) are filtered out. This transformation serves as a basis for efficient processing in the next step.

Second, the coarse biclustering seeds are generated through a partitioning process. The partitioning process constructs a hierarchical tree where all valuable upper level information is kept intact and propagated into the lower level. This helps in preventing any information loss. The binning and partitioning ensures that genes with consistent trends under condition transitions are kept together in the same seeds while genes with inconsistent trends are separated into different seeds. This scheme helps maintain the bicluster quality effectively. Figure 2 shows the partitioning procedure of matrix A''.

Finally, the biclustering seeds are further refined to reflect the similarity of trends' degree of fluctuation. The similarity among the trends is mainly controlled by the same binning and partitioning procedures as those in the previous step. This scheme allows simultaneous grouping of multiple genes and conditions, which makes QHB efficient.

B. Bicluster quality measures

A mean fluctuating degree (MFD) that reflects the similarity of the trend in their degrees of fluctuation is defined in QHB in order to measure the quality of a bicluster. Let pair (X, Y) be a sub-matrix in A', then $MFD(X, Y)$ is defined as:

$$MFD(X,Y) = \sqrt{\frac{1}{|X||Y|} \sum_{x \in X, y \in Y} (A'_{xy} - \frac{1}{|X|} \sum_{x' \in X} A'_{x'y})^2} \tag{1}$$

In a bicluster, if genes have similar degrees of fluctuation in their expression trends under each condition transition, the MFD of the bicluster will be relatively lower. Biclusters that do not satisfy a user specified MFD threshold can be removed.

C. Inter-bicluster Relationship

QHB generates a hierarchical biclustering tree to reveal inter-bicluster relationships. Based on the tree, users can navigate up or down the tree to get a more general or detailed insight into biclusters. The upper part of Figure 1 shows the seed refining process in a hierarchical biclustering tree. The root bicluster is refined further level by level, generating child biclusters with higher degree of similarity in their expression fluctuation trends.

IV. SKB PHASE 2, GENE DISTANCE MEASURE AND CLUSTER GRAPH GENERATION

Clustering and cluster analysis methods are often rooted in a distance scoring function. With a distance function based on domain knowledge and an appropriate threshold, a gene expression bicluster can be converted into an undirected graph, where genes are vertices and functionally similar genes are connected with undirected edges. In general, we consider two genes to be functionally associated if they share at least one biological feature. In this paper, we adopt GO annotation for the gene distance measure. Clearly, this algorithm can be easily extended to adopt other domain knowledge.

A. GO term similarity measure

To model the biological information in different gene sets, we need to take into account that not all the GO terms are equally informative in terms of the biological domain they describe [16]. Therefore, for each gene set, we assign specific weights to the GO terms based on the method as it was done in [17]: the weight of a GO term is defined as the ratio of the number of occurrences (annotations) of the GO term and any of its descendants in the gene set to the total number of term occurrences. For any GO term $t \in T$, T is the full set of GO terms, we define term weight $w(t)$ in Equation 2. By this definition, the root term always has a weight of 1.

$$w(t) = \frac{freq(t) + \sum_{d \in D_t} freq(d)}{N} \quad (2)$$

where $freq(x)$ denotes the number of occurrences of GO term x in a given gene set; D_t is the set of descendants of GO term t in T; and N is the total number of term occurrences.

Given two GO terms t_a and t_b and their corresponding weights $w(t_a)$ and $w(t_b)$, we adopt an enriched GO term comparison method based on the distance to the nearest common ancestor term [18] to assign a term similarity score for t_a and t_b, denoted as $sim(t_a, t_b)$. Since GO allows multiple parents for each term, two terms may share one or more common parents via different paths. We denote the GO term of the lowest common parent as t_{ab}. Then the similarity between GO terms t_a and t_b is defined as:

$$sim(t_a, t_b) = \frac{2 \times \ln w(t_{ab})}{\ln w(t_a) + \ln w(t_b)} \quad (3)$$

where $w(x)$ is the weight of GO term x in T. As $1 \geq w(t_{ab}) \geq w(t_a)$ and $1 \geq w(t_{ab}) \geq w(t_b)$, $sim(t_a, t_b) \in [0, 1]$.

B. Gene distance measure

Biologically, many genes are involved in multiple cellular processes and they are therefore labeled with more than one GO term. For example, the Yeast genome is currently annotated with an average of 6.0 non-IEA GO terms per gene[1]. We define the distance between two gene as the minimum dissimilarity found between any two GO terms that annotate them, as suggested in [19]. Mathematically, let T_{g_i} and T_{g_j} be the set of GO terms annotated to gene g_i and g_j respectively, we define the gene distance measure $d(g_i, g_j)$ as follows:

$$d(g_i, g_j) = \min_{t_a \in T_{g_i}, t_b \in T_{g_j}} ((1 - sim(t_a, t_b)) \times w(t_{ab})) \quad (4)$$

where $sim(t_a, t_b)$ denotes the similarity between GO term t_a and t_b computed with Equation 3, and $w(t_{ab})$ denotes the GO term weight of the lowest common parent of t_a and t_b, which is an adjustment factor using the shared information content to avoid the shallow annotation problem [19]. $d(g_i, g_j) \in [0, 1]$. Note that $d(g_i, g_j)$ is close to 0 as long as there is at least one good GO term match among the lists of GO terms in T_{g_i} and T_{g_j}. In other words, two genes are considered functionally associated if they share at least one biological feature.

C. Cluster graph

For a bicluster S, in order to reveal the biological relationships among the biclustered genes, we convert S into a cluster graph. We define a cluster graph and its component as follows:

Definition 4.1: **Cluster Graph.** A cluster graph $G(V, E)$ is an undirected graph obtained from a bicluster S, such that each vertex in V represents a unique gene in S, and edge $e(g_i, g_j) \in E$ if and only if $d(g_i, g_j) < \sigma$, where $g_i,\ g_j \in V$ and σ is a predefined gene distance threshold, $\sigma > 0$.

Definition 4.2: **Component.** Component $C_i(V_i, E_i)$ is a subgraph of cluster graph $G(V, E)$, such that there exists at least one path between any pair of vertices in V_i, and no path exists between any vertex in V_i and any vertex in $V - V_i$.

Based on these definitions, G is either a connected graph or a set of components, depending on the value of σ. An example of a cluster graph is shown in Figure 11. Note that genes in a component are usually considered to be functionally related. It is also possible that genes in different components are functionally related. In fact, biclustering methods solely based on microarray data would inevitably miss certain function-related genes and thus weaken the overall gene relationship. Therefore, if a key gene is missing from a bicluster, the resulting cluster graph may be broken and the functionally associated genes may be separated into different components.

We observe that the distance value along a path in a cluster graph is possible to be smaller than the direct distance value between the two genes at the ends of the path. In other words, the gene distance measure is intransitive.

Lemma 4.1: **Intransitivity.** *Given two genes g_a and g_b in a cluster graph $G(V, E)$, there may exist a third gene g_x such that $d(g_a, g_b) > d(g_a, g_x) + d(g_b, g_x)$.*

Proof: We prove the existence of such g_x in the following case. Let us consider gene g_a, g_b and g_x in figure 3. $f_1 \dots f_4$ are 4 distinct functions. Gene g_a and g_x share function f_1, and gene g_b and g_x share function f_2, but gene g_a and g_b do not share any function. The gene distance based on Equation 4 are $d(g_a, g_x) = 0$, $d(g_b, g_x) = 0$ and $d(g_a, g_b) = 1$. The distance between g_a and g_b along path "$g_a - g_x - g_b$" is shorter than $d(g_a, g_b)$ (for simplicity, we use 0 and 1 to represent the values of short distance and long distance).□

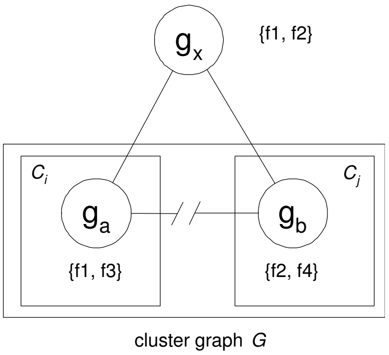


Fig. 3. $f_1 \dots f_4$ are 4 distinct gene functions. Genes g_a and g_x share function f_1, g_b and g_x share function f_2, but g_a and g_b does not share any function. g_a is in component C_i and g_b is in component C_j in a cluster graph G. g_x is not in G but can draw C_i and C_j closer if it is added to G.

Such cluster graph property provides the opportunity to add new genes to the cluster graph to draw its components closer. In the example in Figure 3, if genes g_a and g_b are in two components C_i and C_j in a cluster graph G, and g_x is not in G, the introduction of g_x as a new gene in G may connect C_i and C_j via path "$g_a - g_x - g_b$", if the distance along the path is shorter than the distance threshold σ. The incorporation of such a new gene g_x may help reveal the relationship among the disconnected genes in the cluster graph.

[1] *http://www.geneontology.org/GO.current.annotations.shtml*

In the next section, we introduce the strategies to identify new genes and present an appropriate framework to achieve the goal efficiently.

V. SKB Phase 3, Identifying New Genes to Improve Biological Significance of Biclusters

The biclustering methods solely based on microarray data may miss certain key genes in the biclusters. Reinforcing the biclustered gene functional associations with new genes is an effective way of improving the biological significance of the biclusters. In this section, we first propose a strategy to identify the most proper new genes. Next, we introduce the framework with pseudo codes, followed by addressing the computational challenges. Finally, a graph mining algorithm is provided to efficiently find such genes.

A. Strategy to identify new genes

Using appropriate new genes to reinforce the biclustered gene functional associations is an effective way of refining biological significance of the biclusters. However, incorporating inappropriate genes will adversely affect the biclustered gene associations. Here, we formulated three rules to discover the appropriate new genes.

Rule 1. Distance along new genes bounded by σ. To increase the overall function similarity of a bicluster by drawing its components closer with new genes, the path between any two components along a new gene should not be longer than the predefined gene distance threshold σ and only one new gene is allowed in the path. In the example in Figure 3, g_x is eligible for inclusion into the cluster only if $d(g_a, g_x) + d(g_x, g_b) < \sigma$.

Rule 2. Connect all the connectable components. If any two components can be connected with a set of new genes, one such new gene should be included. Therefore, all the connectable components will be connected. Let I be the whole set of genes in a microarray dataset and S be a bicluster mined from I. Component $C_i(V_i, E_i)$ and $C_j(V_j, E_j)$ are *connectable* if and only if there exists at least one vertex v_x in $I - S$ such that $min\,(d(v_i, v_x) + d(v_j, v_x)) < \sigma$, where $v_i \in V_i$ and $v_j \in V_j$.

Rule 3. Minimum number of new genes. Only the set of genes with the minimum size that satisfy the first two rules will be included. First, we notice that owing to incomplete biological knowledge in GO annotation, introducing too many new gene could cause the bicluster to be biased toward known information. Specifically, if two genes share the same function that is not yet known or annotated, the distance between them will be artificially large. Therefore, we decided to add the minimum number of new genes as a constraint to weaken such bias towards known information. Second, by minimizing the number of new genes to be included, we are maximizing the number of components each new gene connects to. This results in new genes that are more informative. Third, the objective of adding new genes is to understand the relationships among the biclustered genes, and too many new genes may reduce the gene expression coherence.

In summary, the strategy to identify new gene is to search for the smallest set of new genes to connect all the connectable components in a cluster graph with the distance value along the path smaller than σ. Mathematically, given a bicluster S and its cluster graph $G(V, E)$, the new gene set V_x in $I - S$ satisfies two conditions: i) $\forall$ connectable components C_i & $C_j \in G$, $\exists v_x \in V_x$ such that $min(d(v_i, v_x) + d(v_j, v_x)) < \sigma$ ($v_i \in V_i$ and $v_j \in V_j$); ii) V_x is the smallest set satisfying condition i.

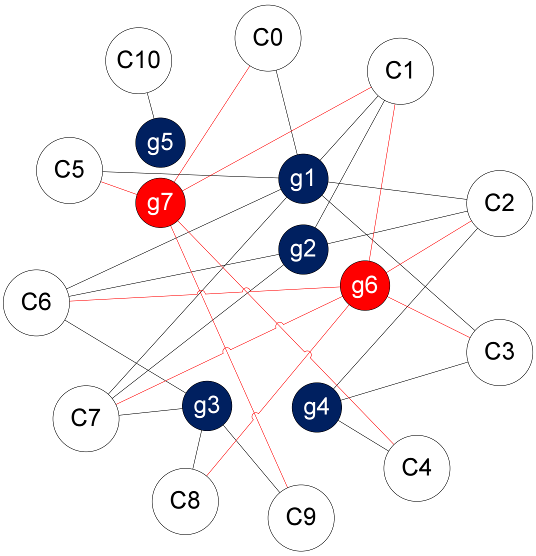


Fig. 4. Illustrative example for new gene identification. $C_0 \ldots C_{10}$ are 11 components in cluster graph G, $g_1 \ldots g_7$ are new gene candidates. Weight of the edge between every component and new gene candidate is set to 0. The red ones are the new genes identified by SKB.

In the illustrative example shown in Figure 4, a cluster graph G has 11 components $C_0 \ldots C_{10}$ and 7 new gene candidates $g_1 \ldots g_7$. For simplicity, the distance value for every edge between a component and a new gene is set to 0. In the example, we aim to find the minimum number of new genes to connect all the connectable components. First, all the new genes satisfy rule 1, since the distance values of all the paths along any new gene are 0. Second, all the components except C_{10} are connectable based on the definition of connectability and hence C_{10} is filtered. Third, we find that g_6, g_7 can connect $C_0 \ldots C_9$, and $g_1 \ldots g_4$'s roles are covered by the selected genes and g_5 cannot bridge any component (see details in Section V-B and V-C). In summary, by adding g_6 and g_7 to G, the number of components is reduced from 11 to 2, *i.e.*, $C_0 + \ldots + C_9 + \{g_6, g_7\}$ and C_{10}, consequently the overall gene functional association in G is much improved.

Note that in this paper we fix the new gene search space to be all the genes in the microarray data, *i.e.*, the root cluster in the hierarchical biclustering tree mined in SKB phase 1. The search space can be easily changed to any ancestor at any level in the hierarchical biclustering tree. Such extension, depending on user's needs, can find new genes that not only satisfy our searching strategy but are also loosely co-expressed with the biclustered genes. In addition, the search space can be easily changed to include genes from the same organism that are not in the microarray data.

B. Algorithm framework

Using GO annotation as domain knowledge, we introduce a new algorithm to identify new genes with the searching strategy illustrated in Section V-A.

As shown in Algorithm 1, the inputs are a gene bicluster S and a user-defined gene distance threshold σ; the outputs of the algorithm are a set of new genes V_x and the resultant cluster graph G'. The proposed algorithm consists of two main steps. First, by computing the gene distances pair-wise for all genes in S with GO annotation, we convert S to a cluster graph $G(V, E)$ with a distance threshold σ (Lines 3-8); and all the connectable components in G are saved in set $\mathbb{C}$ (Lines 9-17). Second, the most appropriate new gene set V_x is introduced to connect all the components in $\mathbb{C}$, resulting in a new cluster graph G' (Lines 18-19). In Algorithm 1, function $Connectable(C_i, C_j)$ returns true if there exists a new gene v_x in $I - S$ such that component C_i can connect to C_j through v_x, otherwise it returns false. Function $getExtraGenes(\mathbb{C})$ returns the smallest new gene set V_x and the corresponding edge set E_x to connect all the connectable components in $\mathbb{C}$.

Algorithm 1 SKB phase 3

```
1:  Input:  $S$ - bicluster;
            $T$ - the full set of GO terms;
            $I$ - the full set of genes in microarray data;
            $\sigma$ - gene distance threshold;
2:  Output: $V_x$ - new gene set;
            $G'$ - the new cluster graph;
3:  $V = \emptyset$; $E = \emptyset$;
4:  for each gene pair $g_i,\ g_j \in S\ (i \neq j)$ do
5:      if $getDistance(g_i, g_j, T) < \sigma$ then
6:          $V = V \cup \{g_i,\ g_j\}$; $E = E \cup \{< g_i, g_j >\}$;
7:      end if
8:  end for
9:  $G = (V, E)$;
10: $\mathbb{C}' = getComponent(G)$;
11: if $|\mathbb{C}'| = 1$ then
12:     $V_x = \emptyset$; $G' = G$;
13: else
14:     $\mathbb{C} = \emptyset$;
15:     for each component pair $C_i,\ C_j \in \mathbb{C}'$ do
16:         if ($Connectable(C_i, C_j)$ = true)
                $\mathbb{C} = \mathbb{C} \cup \{C_i,\ C_j\}$;
17:     end for
18:     $(V_x, E_x) = getExtraGenes(\mathbb{C})$;
19:     $G' = (V \cup V_x, E \cup E_x)$;
20: end if
21: return $V_x$ and $G'$;
```

Note that the orphan vertices in the new cluster graph G' means these genes have distinct functions, and there is no new gene that can link them to other genes in S. They could be removed from the bicluster to increase the overall functional relation. However, owing to the incomplete biological knowledge, we do not consider shrinking the bicluster by removing these orphan vertices.

Having introduced the algorithm framework, we now illustrate the computational difficulties for achieving the goal. In function $getExtraGenes(\mathbb{C})$, a straight-forward method of finding the minimum set of new genes V_x is to test every subset of $I - S$ exhaustively with its size increasing from 1 to $|I - S|$, and the program stops only when the subset of genes can connect all the connectable components in $\mathbb{C}$. The computational time is exponential to the size of the search space $|I - S|$. This leads to a problem in scalability. Because a bicluster S usually has only dozens of genes while I can easily scale to tens of thousands genes, even to the whole genome, such exhaustive search becomes impractical.

C. Finding new genes efficiently

In this section, we introduce an efficient algorithm (see Algorithm 2) to improve the speed in finding the minimum set of new genes V_x, mainly by i) finding a local optimized V_x with a heuristic algorithm, ii) further optimizing the results with a randomized process. The inputs of Algorithm 2 are the gene bicluster S, the connectable component set $\mathbb{C}$, and the full gene set I. The outputs are a set of new genes V_x and the corresponding edges between vertices in V_x and V, denoted as E_x. Algorithm 2 consists of three main steps.

First, for a given bicluster S, instead of testing every new gene candidate in $I - S$, we reduce the search space by considering only the genes that can connect at least two components, denoted as Λ. Other genes are filtered because they can never connect multiple components. We also compute the number of components that g can connect, denoted as $N(g)$ (lines 3-7). The running time is linear to $|I - S|$.

Second, for a new gene candidate set Λ, we compute the upper bound κ of the new gene set size with a locally optimized algorithm by sorting the candidates based on $N(g)$ and finding the new genes serially. In detail, we iteratively move gene g from Λ to V_x if g can connect to the most number of components in $\mathbb{C}$ (lines 10-11). With g, we compose a new component C_x and the number of components is reduced to $|\mathbb{C}| - N(g) + 1$ (lines 12-17). Then we recompute the $N(g)$ for all the genes in Λ (lines 18-21). The iteration stops when Λ is empty. The running time is linear to $|\Lambda|$. In Algorithm 2, function $getConnComp(g, \mathbb{C}, \sigma)$ returns the number of the components in $\mathbb{C}$ under the condition that the distances to their partners through g is smaller than σ. Function $getMinDist(C_i, g)$ returns gene g_i in C_i, such that for all genes in C_i, the distance between g_i and g is the smallest.

Third, we iteratively call a randomized process $FastExtraGene(\Lambda, \kappa)$ to identify the new genes (lines 25-28). In the iteration, if the size of the new gene search space is reduced to be smaller than a constant value MIN_SIZE (in our experiment, 20), we call the exhaustive search to identify the new genes (details in Algorithm 3). Otherwise, as long as the search space is large, the probability of randomly hitting the right new genes is pretty small (see Lemma **??**). Hence, we randomly remove $|\Lambda|/(2 \times \kappa)$ candidates (details in Algorithm 4) and perform two recursive calls. In the following text, we prove that the time complexity of this step is $O(|\Lambda| \times \log^3 |\Lambda|)$.

Lemma 5.1: **Time complexity.** *The running time of* $FastExtraGene(\Lambda, \kappa)$ *is* $O(|\Lambda| \times \log |\Lambda|)$.

Proof: Let T(x) be the time complexity of $FastExtraGene$ for x candidates. In Algorithm 3, the two calls to $RandRemove(\Lambda, t)$ take $O(2 \times |\Lambda|/(2 \times \kappa))$ time; and the two recursive calls on the resulting candidate sets take $2 \times T(|\Lambda| - |\Lambda|/(2 \times \kappa))$ time. Hence, we have $T(|\Lambda|) = O(|\Lambda|/\kappa) + 2 \times T(|\Lambda| - |\Lambda|/(2 \times \kappa))$. Since $T(|\Lambda|) >> \kappa$, we have $T(|\Lambda|) \sim O(|\Lambda|) + 2 \times T(|\Lambda|/2)$. By solving the formula, we get the time complexity $O(|\Lambda| \times \log |\Lambda|)$.□

Algorithm 2 $getExtraGenes(\mathbb{C})$

```
Input:  $S$ - bicluster;
        $T$ - the full set of GO terms;
        $I$ - the full set of genes in microarray data;
        $\sigma$ - gene distance threshold;
        $\mathbb{C}$ - the set of connectable components;
        $c$ - a constant large number;
Output: $V_x$ - new gene set;
        $E_x$ - edge set that connect vertices in $V_x \& V$;
$V_x = \emptyset$; $E_x = \emptyset$; $\Lambda = \emptyset$; $C_x = \emptyset$;
for each gene $g \in I - S$ do
    $N(g) = getConnComp(g, \mathbb{C}, \sigma)$;
    if $(N(g) \geq 2)$ $\Lambda = \Lambda \cup \{g\}$;
end for
$\Lambda' = \Lambda$;
while $\Lambda \neq \emptyset$ do
    remove $g$ with max $N(g)$ from $\Lambda$;
    $V_x = V_x \cup \{g\}$; $C_x = C_x \cup \{g\}$;
    for each component $C_i \in \mathbb{C}$ that $g$ connects to do
        $g_i = getMinDist(C_i, g)$;
        $E_x = E_x \cup \{< g, g_i >\}$;
        $C_x = C_x \cup C_i$; remove $C_i$ from $\mathbb{C}$;
    end for
    $\mathbb{C} = \mathbb{C} \cup \{C_x\}$; $C_x = \emptyset$;
    for each gene $g \in \Lambda$ do
        $N(g) = getConnComp(g, \mathbb{C}, \sigma)$;
        if $(N(g) < 2)$ remove $g$ from $\Lambda$;
    end for
end while
$\kappa = |V_x|$;
if $\kappa > 2$ then
    for $(i = 0;\ i < c \times \log^2 |\Lambda'|;\ i{+}{+})$ do
        $(V'_x, E'_x) = FastExtraGene(\Lambda', \kappa)$;
        if $(|V_x| > |V'_x|)$ $V_x = V'_x$; $E_x = E'_x$;
    end for
end if
return $V_x$ and $E_x$;
```

Algorithm 3 $FastExtraGene(\Lambda, \kappa)$

```
Input:  $\Lambda$ - the new gene candidate set;
        $\kappa$ - new gene set upper bound;
        $\sigma$ - gene distance threshold;
        $\mathbb{C}$ - the set of connectable components;
Output: $V'_x$ - new gene set;
        $E'_x$ - edge set that connect vertices in $V'_x \& V$;
if $|\Lambda| < MIN\_SIZE$ then
    $(V'_x, E'_x) = callExhaustiveSearch(\Lambda, \mathbb{C}, \sigma)$;
else
    $\Lambda_1 = RandRemove(\Lambda, |\Lambda|/(2 \times \kappa)$;
    $\Lambda_2 = RandRemove(\Lambda, |\Lambda|/(2 \times \kappa)$;
    $(V_{x1}, E_{x1}) = FastExtraGene(\Lambda_1, \kappa)$;
    $(V_{x2}, E_{x2}) = FastExtraGene(\Lambda_2, \kappa)$;
    $V'_x = min(V_{x1}, V_{x2})$;
    $E'_x$ = the corresponding edge set of $V'_x$;
end if
return $V'_x$ and $E'_x$;
```

Algorithm 4 $RandRemove(\Lambda, t)$

```
Input:  $\Lambda$ - the new gene candidate set;
        $t$ - number of the randomly removed candidates;
Output: $\Lambda$ - the resultant new gene candidate set;
for $(i = 0;\ i < t;\ i{+}{+})$ do
    randomly select a gene $g \in \Lambda$;
    remove $g$ from $\Lambda$;
end for
return $\Lambda$;
```

TABLE IV

VALUE OF $N(g)$ IN THE EXAMPLE FOR NEW GENE IDENTIFICATION

New gene candidate	**The value of N(g)**			
	Iter 1	**Iter 2**	**Iter 3**	**Iter 4**
g_1	7 ✓	-	-	-
g_2	6	2	1	-
g_3	4	3 ✓	-	-
g_4	4	2	2 ✓	-
g_5	1	-	-	-
g_6	6	2	1	-
g_7	5	3	2	-

In the illustrative example shown in Figure 4, we first filter g_5 since it cannot bridge any component. Next, we obtain the upper bound of new gene set size by compute the values of $N(g)$ for all of the candidates (shown in Table IV) and select new genes serially. According to Algorithm 2, g_1 is firstly selected because it can connect the most number of the components, *i.e.*, 7. In the second iteration, the values of $N(g)$ for the rest of the candidates are recalculated, shown in Table IV column 3. g_3 and g_7 are able to connect to the most number of the components, thus one of them is selected. In the third iteration, g_4 or g_7 is selected. Hence, a new gene set $\{g_1, g_3, g_4\}$ may be identified, with which, all the connectable components are connected. Other candidate genes are automatically filtered since their roles are covered by the selected genes. Therefore, the upper bound of the new gene set size is 3. Finally, by iteratively calling the randomized process in Algorithm 3 and 4 for a number of times, the final new gene set $\{g_6, g_7\}$ is identified. Clearly, $\{g_6, g_7\}$ is better than $\{g_1, g_3, g_4\}$ according to the searching strategy. Note that, in this example, such new gene set cannot be captured by selecting new genes serially, because g_1 is always selected by the serial process in the first iteration, given $N(g_1) > N(g_6) > N(g_7)$.

In summary, all the three steps of SKB phase 3 have polynomial time complexity. In the experiment section below, we investigate the accuracy and efficiency of SKB with different number of new genes and different size of new gene search space respectively.

VI. Experiments

To test the performance of SKB, we implemented SKB in C++ and performed experiments on two datasets, the Yeast cell cycle microarray data[2] in [4] and the Arabidopsis cold-response microarray data collected from NASC, NCBI, TAIR websites[3] and [11], with two GO categories, molecular function and biological process, adopted as domain knowledge.

The Yeast microarray dataset contains 2884 Yeast genes whose expression is altered during cell cycle under 17 time points [4]. Among these genes, 1649 have at least one molecular function annotation, and 1973 have at least one biological process annotation.

The Arabidopsis microarray dataset contains 2255 Arabidopsis cold-response genes under 14 time points with cold treatment at 4°C. Among these genes, 1142 have at least one molecular function annotation, and 975 have at least one biological process annotation.

A. Experimental results

We tested SKB on Yeast and Arabidopsis datasets. The results for all the three phases of SKB are shown below.

In SKB phase 1 of hierarchical biclustering, we adopted a recent top-down hierarchical biclustering algorithm QHB [8] to mine biclusters and inter-cluster relationships. For the Yeast dataset, 109 biclusters were mined, and each bicluster had on average 56.2 genes. The hierarchical biclustering tree contained 6805 nodes, including 3329 seeds and 109 leaves. Out of the 2884 cell cycle responsive genes, 1133 were in at least one bicluster. For the Arabidopsis dataset, 59 biclusters were mined, and each bicluster had on average 35.2 genes. The hierarchical biclustering tree contained 2123 nodes, including 1178 seeds and 59 leaves. Out of the 2255 cold-response genes, 374 were in at least one bicluster. By drawing the hierarchical biclustering trees, the inter-cluster relationships are revealed.

In SKB phase 2 of cluster graph generation, we set distance thresholds σ for each microarray dataset under different GO categories, such that for any edge in cluster graph $G(V, E)$, the false discovery rate (FDR) [20] of its distance value is smaller than 0.01. Hence all the edges in all of the cluster graphs are statistically significant. The values of σ used in our experiment are shown in Table V. With the given distance thresholds, we convert all of the 109 Yeast biclusters and 59 Arabidopsis biclusters into cluster graphs. Note that all of the un-annotated genes are removed from the cluster graphs before further processing.

In SKB phase 3 of new gene identification, a set of new genes are retrieved for each bicluster. On average, for Yeast biclusters, 7.3 and 9.6 new genes were found per cluster with molecular function and biological process annotations respectively. For Arabidopsis biclusters, 1.8 and 3.2 new genes were found per cluster with molecular function and biological process annotations respectively (see details in Table V).

B. Algorithm performance evaluation

To qualitatively show the biological significance of the biclusters mined with SKB, we introduce two measures: the Graph Connectivity Score (GCS) and the Overall functional Similarity Score (OSS).

GCS is defined in Equation 5, which tests the overall functional association of a bicluster from a topology perspective.

$$GCS(G) = \frac{\sum_{i=1}^{n} |C_i|^2}{\left(\sum_{i=1}^{n} |C_i|\right)^2} \tag{5}$$

where n is the number of components in G. If the value of $GCS(G)$ is close to 1, most gene pairs are connectable through at least one path in G. In contrast, the value of $GCS(G)$ close to 0 signifies that most of the genes are not reachable to other genes.

OSS is defined in Equation 6, which tests the overall functional similarity of a bicluster using GO annotation. The value close to 1 means that, for most of the genes in cluster graph G, the overall pair-wise similarity of the connected is high, or the similarity value along the path to connect the gene pair is high. In contrast, the value of $OSS(G)$ close to 0 means most of the genes are dissimilar from each other.

$$OSS(G) = 1 - \frac{\sum_{i,j=1}^{|V|} f(g_i, g_j)}{\frac{1}{2} \times |V| \times (|V| - 1)} \tag{6}$$

$$f(g_i, g_j) = \begin{cases} d(g_i, g_j) & \text{if } < g_i, g_j > \in E \\ \min\limits_{\phi \in \Phi(g_i, g_j)} \sum\limits_{(u,v) \in \phi} d(u, v) & \text{if } < g_i, g_j > \notin E \& \Phi \neq \emptyset \\ 1 & \text{otherwise} \end{cases}$$

where $\Phi(g_i, g_j)$ denotes the set of paths connecting g_i and g_j. $f(g_i, g_j)$ is the gene distance value if g_i and g_j are directly connected, or the sum of distance along the shortest path connecting g_i and g_j if such path exists. If no such connections exist between two genes, the value is 1.

We evaluated the performance of SKB by comparing the GCS and OSS scores with those in the biclusters mined with QHB. We chose QHB to compare because i) QHB outperforms other existing biclustering methods in mining biclusters with consistent trends and trends with similar degrees of fluctuations [8]; ii) such comparison shows the effectiveness of the new genes since SKB generates the same biclusters as QHB. To evaluate SKB's strategy to identify new genes, we also compared the GCS and OSS scores of SKB with those in a random selection process, in which we randomly choose the same number of new genes as SKB does from Λ and add them to each bicluster.

[2] *http://arep.med.harvard.edu/biclustering*

[3] *http://affymetrix.arabidopsis.info/narrays/experimentbrowse.pl; http://www.ncbi.nlm.nih.gov; http://www.arabidopsis.org*

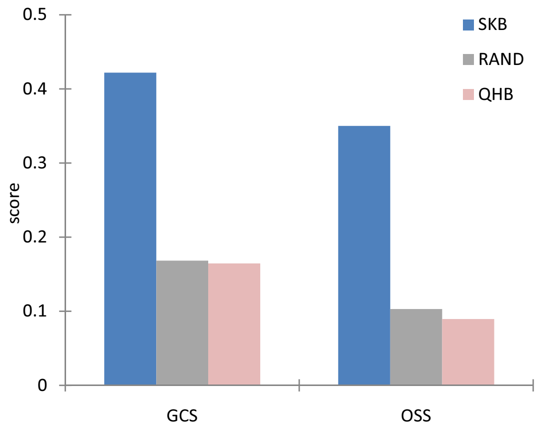


Fig. 5. GCS and OSS scores of Arabidopsis clod-related gene biclusters improved with biological process annotations.

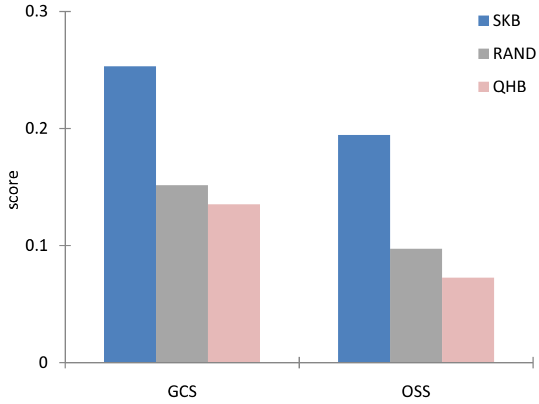


Fig. 6. GCS and OSS scores of Arabidopsis clod-related gene biclusters improved with molecular function annotations.

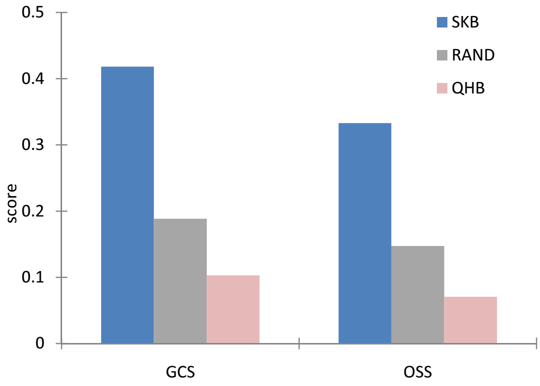


Fig. 7. GCS and OSS scores of Yeast cell-cycle gene biclusters improved with biological process annotations.

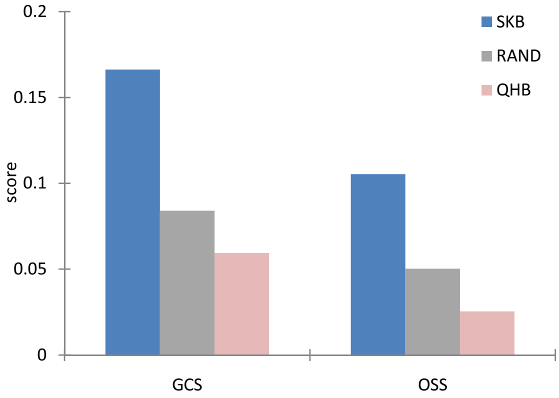


Fig. 8. GCS and OSS scores of Yeast cell-cycle gene biclusters improved with molecular function annotations.

Experimental results shown in Table V indicate that the GCS and OSS scores of the biclusters are considerably increased comparing to QHB and the random selection. Figure 5 shows that by introducing an average of 3.2 new genes to the Arabidopsis biclusters with biological process annotation, comparing to QHB and the random selection, the mean values of GCS and OSS increase 156.4% and 291.1%, 150.7% and 239.8% respectively. Figure 6 shows that by introducing an average of 1.8 new genes to the Arabidopsis biclusters with molecular function annotation, comparing to QHB and the random selection, the mean values of GCS and OSS increase 87.4% and 167.8%, 67.1% and 99.7% respectively.

Figure 7 shows that by introducing an average of 9.6 new genes to the Yeast biclusters with biological process annotation, comparing to QHB and the random selection, the mean values of GCS and OSS increase 305.9% and 371.0%, 121.9% and 126.2% respectively. Figure 8 shows that by introducing an average of 7.3 new genes to the Yeast biclusters with molecular function annotation, comparing to QHB and the random selection, the mean values of GCS and OSS increase 179.6% and 315.1%, 97.1% and 109.7% respectively.

To test the efficiency of SKB, we compared SKB with the exhaustive search on the Arabidopsis dataset using biological process annotations on a 3.0GHz Pentium PC with 1.5GB memory. Figure 9 shows that SKB is much faster than the exhaustive search, with 100- to 100k-fold speed up (time for the exhaustive search for over 1000 genes are putatively calculated). Similar trends were shown in Yeast dataset or using molecular function annotations. We further study the accuracy of the new genes identified by SKB. Due to the impractically computational time it takes for the exhaustive search, we analyzed the accuracy by investigating whether there exists a smaller new gene set satisfying the searching strategy, given a bicluster S and the set of new genes V_x identified by SKB. In our experiment, we repeatedly and randomly choose $|V_x|-1$ genes from Λ for 1 million times. If none of the randomly selected gene sets satisfies the searching strategy, we say SKB is accurate for bicluster S, otherwise SKB is inaccurate. Experiment shows that SKB can always find the right new gene set V_x for its size ranging from 1 to 10 for both of the experimental datasets.

In Figure 10, the functional characterization of the new genes for Arabidopsis shows that there is an enrichment of transcription factors (TF) in the new genes compared to all the genes in the microarray data. TFs are regulate transcriptions and are generally located in nucleus. A TF usually does not co-express substantially with its target genes. Hence, such TFs are usually missing from biclusters based on expression data. The significant and consistent rise of transcription factors in

TABLE V

SKB PERFORMANCE SUMMARY

Genome	GO Category	Distance threshold	# New Genes		GCS increasement		OSS increasement	
			Overall	per cluster	v.s. QHB	v.s. Rand	v.s. QHB	v.s. Rand
Arabidopsis	Biological process	0.122	35	3.2	156.4%	150.7%	291.1%	239.8%
	Molecular function	0.136	13	1.8	87.4%	67.1%	167.8%	99.7%
Yeast	Biological process	0.159	286	9.6	305.9%	121.9%	371.0%	126.2%
	Molecular function	0.232	168	7.3	179.6%	97.1%	315.1%	109.7%

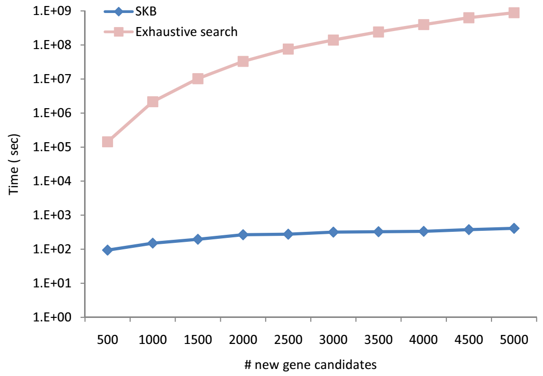


Fig. 9. Comparison of computational times to find new genes in the Arabidopsis dataset using biological process annotation. It shows that SKB is 100- to 100k-fold speedup.

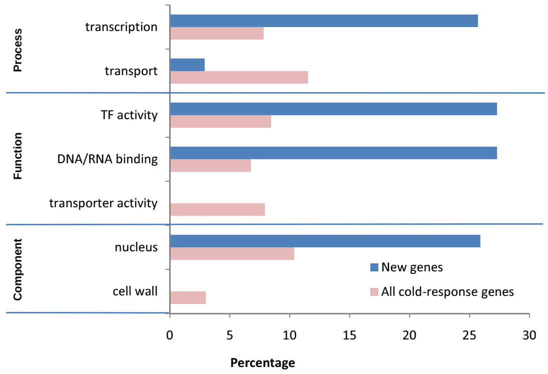


Fig. 10. The functional characterization distribution of the new genes comparing to all the genes in the microarray data. It shows that the new genes has more transcription factors.

the new gene set from the highly noisy background on all the three GO categories indicates that SKB is biologically sound.

C. Arguments

It may be argued that instead of setting a sophisticated algorithm to add new genes, a connected cluster graph can be easily obtained by simply increasing the distance threshold. To address this question, we increased the distance thresholds until the same number of components as those connected by SKB was reached. To reach this point, the distance threshold was increased on average from 0.122 to 0.665 for the Arabidopsis dataset with biological process annotations. Consequently, the resultant cluster graphs are cliques or clique-like dense graphs, in which the edges are not statistically significant and numerous weak gene associations are present. By contrast, the distance threshold in SKB is constantly low and all the edges are statistically significant ($FDR < 0.01$). Therefore, SKB is significantly better on graph representation and noise control than simply increasing the distance threshold.

It may also be argued that the effort to introduce new gene can be replaced by simply joining two related biclusters because the biclusters are highly overlapped and new genes found for one bicluster may already exist in another bicluster. To address this question, we tested how many new genes identified by SKB were already included in the biclusters. For Arabidopsis, only 12 out of 35 and 3 out of 13 new genes are overlapped with biclustered genes using biological process and molecular function annotation respectively. This indicates that the majority of the new genes cannot be captured by joining related biclusters. Therefore, SKB is more effective than joining existing clusters because it finds considerably more new genes.

D. Case study

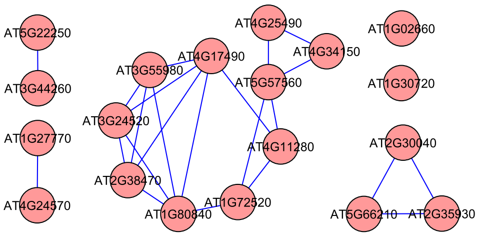


Fig. 11. The cluster graph of Arabidopsis cold-response bicluster #800 has 6 components.

A case study on Arabidopsis data set is shown in Figure 11 and 12. Arabidopsis bicluster #800 has 19 annotated genes. Using biological process annotation, the cluster graph is formed with one large component, denoted as C_1, a triangle, denoted as C_2, and the other four small components. C_1 belongs to the *defense response* process and C_2 belongs to the *protein amino acid phosphorylation* process. These two components are disconnected because they are conceptually dissimilar based on GO annotation. To study why the two

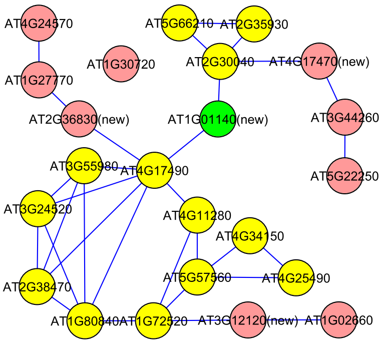


Fig. 12. SKB introduces 4 new gene to connect the genes in bicluster #800, in which, AT1G01140 (green colored) bridges the two yellow colored components.

different processes co-express under cold treatment and how these processes may work together, SKB introduces 4 new genes that can connect the biclustered genes, in which, AT1G01140 (green colored) bridges C_1 and C_2. AT1G01140 is a signal transduction receptor. In general, a signal transduction receptor is the first component an organism uses to respond to a signal. In the two components C_1 and C_2, C_1 may contain genes that respond to a pathogen and C_2 may contain genes that process the signal from the pathogen, but we do not know if they are responding to the same signal. Because the new gene, which is a signal receptor, connects both of these graphs, we can hypothesize that the two processes may work together in responding to the pathogen via that signal receptor.

This case study supports that, by focusing on mining gene relationships, SKB can recognize functional modules in biclusters and identify interesting new genes to bridge the functionally distinct biclustered genes, which will considerably enhance the interpretability of microarray data.

VII. Conclusion

In this paper, we introduce a biclustering method SKB for gene expression clustering. SKB has three phases. First, a hierarchical biclustering method is adopted to mine biclusters from microarray data and discover the inter-cluster relationships. Second, biclusters are converted into cluster graphs using a GO annotation based distance measure. Third, additional genes, if there are any, are identified to re-connect the components in each cluster graph, in order to uncover the otherwise hidden gene associations within biclusters. The delineation of functional relationships and incorporation of such missing genes may help biologists to discover biological processes that are important in a given study and provide clues for how the processes may function together. Experimental results show that SKB can reveal the inter- and intra-bicluster relationships efficiently and accurately, and greatly improve the biological significance of the biclusters.

We will extend the capability of SKB in two ways in the future. First, heterogeneous domain knowledge, including protein motifs, cis-elements, sequence alignment and others, will be utilized to improve the bicluster performance in addition to the GO annotations. Second, SKB will be further extended to enhance the performance of a gene search engine by grouping functionally similar genes, recommending additional genes that are not in the search result, and visualizing the search result graphically.

References

[1] J. Jun, S. Chung, and D. McLeod, “Subspace clustering of microarray data based on domain transformation,” *VLDB Workshop on Data Mining on Bioinformatics*, 2006.

[2] R. Shamir and R. Sharan, “Click: A clustering algorithm for gene expression analysis,” *ISMB*, 2000.

[3] M. Eisen, P. Spellman, P. Brown, and D. Botstein, “Cluster analysis and display of genome-wide expression patterns,” *PNAS*, pp. 14 863–14 868, 1998.

[4] Y. Cheng and G. Church, “Biclustering of expression data,” *ISMB*, pp. 93–103, 2000.

[5] H. Wang, W. Wang, J. Yang, and P. Yu, “Clustering by pattern similarity in large data sets,” *SIGMOD*, pp. 394–405, 2002.

[6] J. Yang, W.Wang, H.Wang, and P. Yu., “δ-clusters: Capturing subspace correlation in a large data set,” *ICDE*, pp. 517–528, 2002.

[7] Z. Zhang, M. Teo, B. Ooi, and K. Tan, “Mining deterministic biclusters in gene expresssion data,” *BIBE*, pp. 283–290, 2004.

[8] L. Ji, K. Mock, and K. Tan, “Quick hierarchical biclustering on microarray gene expression data,” *BIBE*, pp. 110–120, 2006.

[9] Y. Lazebnik, “Can a biologist fix a radio?–or, what i learned while studying apoptosis,” *Cancer Cell*, vol. 2, no. 3, pp. 179–182, 2002.

[10] M. Ashburner, C. Ball, J. Blake, *et al.*, “Gene ontology: tool for the unification of biology,” *Nat Genet*, vol. 25, no. 1, pp. 25–29, 2000.

[11] J. Vogel, D. Zarka, H. Van Buskirk, *et al.*, “Roles of the cbf2 and zat12 transcription factors in configuring the low temperature transcriptome of arabidopsis,” *Plant Journal*, vol. 41, pp. 195–211, 2005.

[12] W. Pan, “Incorporating gene functions as priors in model-based clustering of microarray gene expression data,” *Bioinformatics*, vol. 22, no. 7, pp. 795–801, 2006.

[13] M. Robinson, J. Grigull, N. Mohammad, and T. Hughes, “Funspec: a web-based cluster interpreter for yeast,” *BMC Bioinformatics*, vol. 3, no. 35, 2002.

[14] F. Sohler, D. Hanisch, and R. Zimmer, “New methods for joint analysis of biological networks and expression data,” *Bioinformatics*, vol. 20, no. 10, pp. 1517–1521, 2004.

[15] A. Prelic, S. Bleuler, P. Zimmermann, *et al.*, “A systematic comparison and evaluation of biclustering methods for gene expression data,” *Bioinformatics*, vol. 22, no. 9, pp. 1122–1129, 2006.

[16] S. Y. Rhee, V. Wood, K. Dolinski, and S. Draghici, “Use and misuse of the gene ontology annotations,” *Nature Reviews Genetics*, 2008.

[17] P. Lord, R. Stevens, *et al.*, “Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation,” *Bioinformatics*, vol. 19, no. 10, pp. 1275–1283, 2002.

[18] D. Lin, “An information-theoretic definition of similarity,” *ICML*, pp. 296–304, 1998.

[19] J. Sevilla, V. Segura, *et al.*, “Correlation between gene expression and go semantic similarity,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 2, no. 4, pp. 330–338, 2005.

[20] Y. Benjamini and Y. Hochberg, “Controlling the false discovery rate: a practical and powerful approach to multiple testing,” *J. Roy Stat Soc., SerB.*, vol. 57, no. 289-300, 1995.