Applied Mathematics & Information Sciences
An International Journal

Efficient Mining Differential Co-Expression Constant Row Bicluster in Real-Valued Gene Expression Datasets

Miao Wang[1]* *and Xuequn Shang*[1] *and Xiaoyuan Li*[1] *and Zhanhuai Li*[1] *and Wenbin Liu*[2]

[1] School of Computer Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China
[2] Department of Physics and Electronic Information Engineering, Wenzhou University, Wenzhou 325035, China

Received: Jul 8, 2012; Revised Oct. 4, 2012; Accepted Oct. 6, 2012
Published online: 1 Mar. 2013

Abstract: Biclustering aims to mine a number of co-expressed genes under a set of experimental conditions in gene expression dataset. Recently, differential co-expression biclustering approach has been used to identify class-specific biclusters between two gene expression datasets. However, it cannot handle differential co-expression constant row biclusters efficiently in real-valued datasets. In this paper, we propose an algorithm, *DRCluster*, to identify Differential co-expression constant Row biCluster in two real-valued gene expression datasets. Firstly, *DRCluster* infers the differential co-expressed genes from each pair of samples in two real-valued gene expression datasets, and constructs a differential weighted undirected sample-sample relational graph. Secondly, the differential co-expression constant row biclusters are produced in the above differential weighted undirected sample-sample relational graph. We also design several pruning techniques for mining maximal differential co-expression constant row biclusters without candidate maintenance. The experimental results show our algorithm is more efficient than existing one. The performance of *DRCluster* is evaluated by MSE score and Gene Ontology, the results show our algorithm can find more significant and biological differential biclusters than traditional algorithm.

Keywords: differential co-expression, biclustering, constant row, gene expression.

1. Introduction

Biclustering [1] is one of the popular methods for gene expression dataset analysis. It can identify a group of co-expressed genes under a subset of experimental conditions. There have existed many biclustering methods. Such as, [1] proposed the first biclustering algorithm which employed a greedy node deletion approach using a low mean squared residue for mining constant value and constant row or column biclusters, [2] focuses on discovering constant value biclusters, coherent evolution biclusters can be found by [3], [4] uses a weighted multi-graph to mine scaling biclusters and [5] proposed to mine biclusters in discretized gene expression dataset. However, above algorithms do not consider the class labels to infer the differential co-expression biclusters in two or more labeled gene expression datasets.

Differential co-expression biclustering methods are used to detect differential co-expression bicluster which shows highly corrected co-expression in one dataset but not in the other. Mining differential co-expression bicluster is more useful for disease detection. For example, it can lead to the identification of age-related genes under age-associated conditions by comparing the expression of the genes between two age periods. Biologically speaking, using differential co-expression bicluster can indicate the wrong regulation of a pathway [6].

Recently, many approaches have been proposed to infer differential co-expression biclusters. [7] used two-steps approach to produce differential co-expression biclusters. It identifies biclusters in each class separately firstly, then above identified biclusters are ranked based on their difference between the two classes. [8] also used two-steps to infer differential co-expression biclusters. The first step is to mine the up or the down regulated genes, then [9] is used to identify the biclusters from the up-regulation and the down-regulation data. Above two-steps procedure for mining differential co-expression bicluster is naive. The produced bicluster in one class may also be inferred in the

* Corresponding author: e-mail: riyushui@gmail.com

other class, which influences the mining efficiency. Therefore, [10] developed a methodology for differential co-expression network analysis for the comparison of gene co-expression on a global scale. Each edge in the difference network represents the change in correlation that occurs between two gene expression classes. Then they identified a number of functional gene groups that change co-expression between two age classes. [5] produces discriminative bicluster in the weighted undirected sample relational graph which is constructed based on difference between two gene expression datasets. Above two methods construct a difference matrix to mine discriminative bicluster. However, differential bicluster may not be differential co-expression bicluster.

The recent proposed *DiBiCLUS* [11] algorithm aims to mine differential co-expression biclusters from two discretized gene expression datasets. Firstly, *DiBiCLUS* identifies the differential pairs of genes. Then the differential biclusters are generated using clustering method. However, there existed some drawbacks of *DiBiCLUS*. Firstly, *DiBiCLUS* can only handle one relation between two genes, which may omit some biological information. Secondly, *DiBiCLUS* cannot be used for mining differential bicluster in real-valued gene expression datasets, which may lose some biological interesting results. Thirdly, *DiBiCLUS* needs to be double mining differential co-expression bicluster. One is to mine differential co-expression biclusters from *Class A* to *Class B*. The other is from *Class B* to *Class A*. Such double checking procedure influences the mining efficiency. Finally, *DiBiCLUS* maintenances the whole cluster in memory, which influences the memory efficiency.

SDC algorithm [12] is another method for mining subspace differential co-expression (SDC) patterns. It can also be used for discovering differential co-expression bicluster. *SDC* algorithm aims to infer the patterns which are co-expressed over a large percent of the conditions in one microarray dataset, but in a much smaller percent of conditions in the other microarray dataset. Unlike some differential co-expression biclustering algorithms which mine DC biclusters in discretized datasets, *SDC* can use the novel range support to produce constant row SDC bicluster in real-valued gene expression datasets. Range support is proposed by [13] which uses it to mine the meaningful patterns which are coherent for a substantial fraction of transactions or samples in the dataset. However, there existed some limitations of range support measure. Firstly, a condition of the range support pattern can only contribute to the range support of gene set if the values of all the genes in it are all positive or negative. Therefore, range support cannot measure the values of some genes in one transaction are not same sign. Secondly, range support can measure coherent genes under one condition, but it cannot handle the coherent conditions of one gene, which is very important for mining bicluster. Finally, range support can only mine coherent genes, but it cannot illustrate the co-expression types between a pair of genes in real-valued dataset.

However, *SDC* framework has some limitations for mining differential co-expression constant row biclusters in real-valued gene expression datasets. Firstly, it adopts the *Apriori* framework which limits its efficiency and scalability. Secondly, the similar to *DiBiCLUS*, *SDC* algorithm also needs to be double mining SDC patterns. One is to mine SDC patterns from *Class A* to *Class B*, the other is to infer from *Class B* to *Class A*. Thirdly, due to the striction of subspace differential co-expression support [13], it may generate very small genes. Fourthly, when mining constant row bicluster using gene-growth procedure, it needs to compute all the coherent samples under each gene. Such two-steps approach is very time-consuming. Finally, based on the definition of subspace differential co-expression, *SDC* cannot find some interesting differential co-expression biclusters.

In hopes of overcoming the limitations of existed differential biclustering methods, we propose an efficient algorithm, *DRCluster*, for inferring Differential co-expression constant Row biClusters in two real-valued gene expression datasets. In order to escape of traditional double-checking approach, our algorithm can produce differential biclusters in one time. Firstly, we infer the differential co-expressed genes in each pair of samples in two real-valued gene expression datasets, and construct a differential weighted undirected sample-sample relational graph. In order to handle the coherent conditions or samples in one gene, we defined a new range support measure, *sample range support*, to measure user-defined coherent meaningful samples. We also defined the three co-expression relationships between a pair of genes in real-valued dataset for mining differential co-expression bicluster in real-valued microarray datasets. Secondly, the differential co-expression constant row biclusters are produced in the above differential weighted undirected sample-sample relational graph. We design several pruning techniques to improve efficiency for generating maximal biclusters without candidate maintenance. The overview of our approach is illustrated in Fig.1.

The contributions of our *DRCluster* framework which distinguish from existing ones are summarized as follows:

(1) *DRCluster* can identify new type of differential co-expression constant row biclusters in two real-valued gene expression datasets.

(2) We define the sample range support to measure coherent samples under genes.

(3) The three types of co-expression relationship between two genes are defined in real-valued gene expression data.

(4) The proposed *DRCluster* algorithm can mine maximal differential co-expression constant row biclusters without candidate maintenance.

2. PRELIMINARIES AND DEFINITIONS

The gene expression data is denoted as $D = G \times C$, where the column C represents the set of experimental condi-

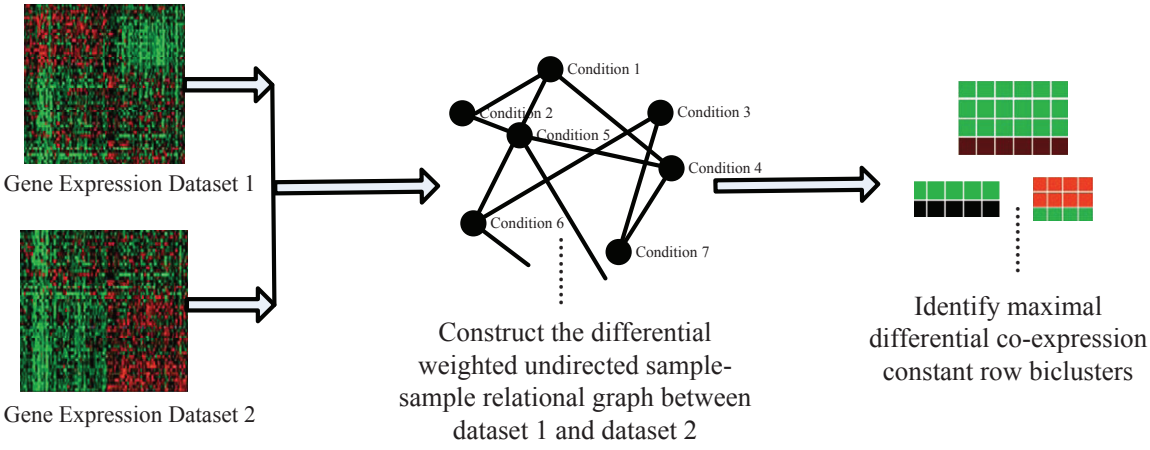

Figure 1 The overview of DRCluster for inferring maximal differential co-expression biclusters.

tions, and the row G represents genes. The element value of D_{ij} is a real value which is the expression level of gene i under condition j. A bicluster P is defined as a sub-matrix of D, denoted as $Samples(Genes)$. For simplicity, we denote the gene set of P as $P.Geneset$ and the conditions of P as $P.Sample$. Given two microarray datasets, D_1 and D_2, shown in Table 1 and Table 2, where $D_1 \subseteq D$ and $D_2 \subseteq D$.

Table 1 An example of real-valued gene expression dataset D_1.

	S_1	S_2	S_3	S_4	S_5
G_1	2.1	2.12	2.11	2.1	2.09
G_2	3.3	3.29	3.3	3.31	3.29
G_3	-1.5	-1.53	-1.55	-0.51	-1.53
G_4	-2.62	-2.61	-2.6	0.61	3.7
G_5	5.5	3.5	8.1	2.2	4.51
G_6	1.6	2.5	3.1	4.2	1.91

Table 2 An example of real-valued gene expression dataset D_2.

	S_1	S_2	S_3	S_4	S_5
G_1	2.87	3.2	4.9	2.2	1.21
G_2	2.1	3.1	3.72	4.1	3.41
G_3	1.1	1.9	2.9	3.8	0.02
G_4	1.54	1.55	1.54	2.1	1.53
G_5	5.2	5.21	-0.2	5.19	5.21
G_6	-1.12	-1.1	-1.13	-1.1	-1.11

As mentioned in [13], a bicluster is interesting if all the genes under conditions are co-expressed based on the following range support definition.

Definition 1. *Given a gene expression dataset D and a self-assignment value α, the range support of a real-valued gene set $G = \{g_1, g_2, \cdots, g_k\}$ is defined as*

$$RangeSupport(G) = \sum_{c \in C} rs(c, G)$$

where $rs(c, G)$ is defined as:

$$rs(c,G) = \begin{cases} \min\limits_{g \in G} |D_{c,g}| & if[\forall g \in G, D_{c,g} > 0 \; or \\ & \forall g \in G, D_{c,g} < 0] \; \& \\ & [(\max\limits_{g \in G} D_{c,g} - \min\limits_{g \in G}) \leq \\ & \alpha(\min\limits_{g \in G} |D_{c,g}|)] \\ 0 & otherwise \end{cases} \quad (1)$$

As mentioned in above section, *range support* needs to measure the expression values of genes having all positive or all negative in one transaction and it cannot measure the coherent samples under genes. Therefore, in this paper, we will mine constant row bicluster in real-valued gene expression dataset using the following *sample range support* measure.

Definition 2. *Given a microarray dataset D and a self-assignment value α. Therefore, the Sample Range Support (SRS) of one real-valued gene set $G = \{g_1, g_2, \cdots, g_k\}$ is defined as*

$$SampleRangeSupport(G) = \sum_{g \in G} srs(g, C)$$

where $srs(g, C)$ is defined as:

$$srs(g,C) = \begin{cases} \min\limits_{c\in C} |D_{g,C}| & if[\forall c \in C, (\max\limits_{c\in C} D_{g,c} - \\ & \min\limits_{c\in C} D_{g,c}) \leq \alpha(\min\limits_{c\in C} |D_{g,c}|)] \\ 0 & otherwise \end{cases} \tag{2}$$

Traditional DC bicluster mining methods [11,12] aim to find different co-expression types between any two genes under a set of samples. [12] discovers a set of genes which are co-expressed on a much larger percent of conditions in one dataset compared to the co-expression on any size-2 subset of genes in the other dataset. However, SDC support can produce many little differential co-expression genes which may influence the biological analysis. In order to escape of producing more little scales of differential co-expression biclusters, we produce DC bicluster using the following definition in this paper.

Definition 3. *A bicluster is differential co-expression bicluster if the co-expressed type of relations (positive or negative) between at least two genes under all the conditions in one class is not the same as the co-expression type of relations between the same genes under the same conditions in the other class.*

The definition 3 indicates that differential co-expression bicluster has at least a pair of genes which is positive co-expression in *Class A*, and is negative co-expression or non-expression in *Class B*. Or it is negative co-expression in *Class A*, and is positive co-expression or non-expression in *Class B*. Our goal is to mine all the maximal differential co-expression biclusters in two real-valued microarray datasets. However, using definition 2 can find set of co-expressed genes under some conditions, but it cannot infer the co-expression relationship among genes. Differential biclustering aims to find gene sets that are co-expressed under a subset of conditions in one class but not in the other class. Therefore, how to infer the co-expression relationship between two genes under a subset of conditions is very important for discovering differential constant row biclusters in real-valued gene expression datasets. Three types of relations between genes G_1 and G_2 in two given conditions C_1 and C_2 can be respectively defined as follows.

(1) G_1 and G_2 is positive co-expression which is denoted as $\{G_1G_2\}$ if $[\forall g \in \{G_1, G_2\}|(\max\limits_{c\in\{C_1,C_2\}} D_{g,c} - \min\limits_{c\in\{C_1,C_2\}} D_{g,c}) \leq \alpha(\min\limits_{c\in\{C_1,C_2\}} |D_{g,c}|)]$ and $[\forall c \in \{C_1, C_2\} |(D_{G_1,c} \times D_{G_2,c} > 0)]$;

(2) G_1 and G_2 is negative co-expression which is denoted as $\{G_1 - G_2\}$ if $[\forall g \in \{G_1, G_2\}|(\max\limits_{c\in\{C_1,C_2\}} D_{g,c} - \min\limits_{c\in\{C_1,C_2\}} D_{g,c}) \leq \alpha(\min\limits_{c\in\{C_1,C_2\}} |D_{g,c}|)]$ and $[\forall c \in \{C_1, C_2\} |(D_{G_1,c} \times D_{G_2,c} < 0)]$;

(3) G_1 and G_2 is non-expression if they are not up-expressed or down-expressed.

Our goal is to mine all the maximal differential co-expression biclusters using above gene co-expression relations definition in two real-valued gene expression datasets. In the next section, we will show how our algorithm mining maximal differential co-expression constant row biclusters.

3. THE DRCLUSTER ALGORITHM

In this section, we will present how our algorithm finding maximal differential co-expression biclusters in two real-valued gene expression datasets. The *DRCluster* framework has two steps as following:

1. Producing differential co-expressed genes in each pair of samples in gene expression datasets, and constructing a differential weighted undirected sample-sample relational graph.

2. Mining maximal DC constant row biclusters in the above differential weighted undirected sample-sample relational graph.

A. Construct the Differential Weighted Undirected Sample-Sample Relational Graph

As mentioned in above, traditional algorithms [11,12] to mine maximal DC biclusters needs to be double times checking, which is less efficient and more time consuming. In order to mine DC biclusters efficiently, *DRCluster* generates maximal DC constant row biclusters from the differential weighted undirected sample-sample relational graph (DWUR Graph). [5] also uses the WUR Graph to mine maximal bicluster in gene expression dataset. The difference between WUR Graph in [5] and this paper is that our DWUR Graph contains differential co-expression genes and our DWUR Graph is produced by real-valued gene expression dataset instead of discretized dataset. The definition of DWUR graph is shown as following.

Definition 4. *The DWUR Graph $G = \{E, S, W\}$, each vertex S_i in the graph represents an unique sample, there exists an edge E_{ij} between a pair of samples only if S_i and S_j have co-expressed genes which satisfy a pair of differential co-expression genes' definition in the definition 3 and weighted item set W_{ij} between S_i and S_j samples is the above differential co-expressed genes. For clarity, W_{ij} is denoted as $S_iS_j.Geneset$.*

The merits of our constructed DWUR Graph being more efficient than traditional differential pairs of genes, are shown as following. (1) As we known, the number of genes in gene expression dataset is much greater than the number of samples. Therefore, using sample-growth method is more efficient than gene-growth. However, if the number of genes is less than samples, gene-growth approach may be more efficient. In this paper, we assume that gene expression dataset has more genes than samples. (2) Since it is possible for any a pair of genes to be positive co-expression and negative co-expression under a set of samples. [11] can only handle one relation between two genes, which may

omit some biological information. Our DWUR Graph contains both positive co-expression and negative co-expression among genes. Therefore, our algorithm can mine both positive co-expression genes and negative co-expression genes under the same set of samples.

According to definition 3, the pair of differential co-expressed genes has two parts. One is that a set of genes are co-expressed in *Class A* and have opposite co-expression or non-expression in *Class B*. The other is that it is co-expressed in *Class B* and has opposite co-expression or non-expression in *Class A*. Therefore, the two parts are not same. For clarity, above two parts of differential co-expression genes are denoted as $PGenes$ and $NGenes$, respectively. For example, $S_1S_2.PGenes$ is $G_1G_2 - G_3 - G_4$ and $S_1S_2.NGenes$ is $G_3G_4G_5 - G_6$ between Table 1 and Table 2. Fig.2 shows an example of DWUR Graph which is constructed from Table 1 and Table 2.

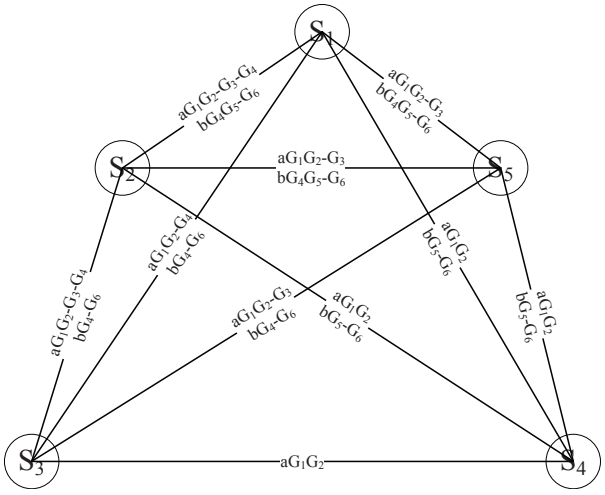

Figure 2 An example of DWUR Graph between Table 1 and Table 2.

B. Mining Maximal Differential Co-expression Constant Row Biclusters

In this section, we will show how our method finding the maximal DC biclusters without candidate maintenance in real-valued datasets. As mentioned above, our algorithm infers biclusters in DWUR Graph, which is constructed by merging the two original microarray datasets according to definition 3. Since the total of conditions is far less than the number of genes, so we generate DC biclusters by using sample-growth (also called condition-growth) method. Before the *DRCluster* algorithm is presented, let's discuss the anti-monotonic of *sample range support*.

Theorem 1. *The sample range support measure is anti-monotonic.*

Theorem 1 states that our sample range support satisfies the anti-monotonic property, which guarantees that we can use *Apriori*-like efficient pattern mining framework to discover biclusters. Therefore, our *DRCluster* algorithm adopts *Aprioir*-like procedure to produce differential co-expression constant row bicluster using sample-growth in DWUR Graph. The following lemma can guarantee that sample-growth is satisfied the range support definition.

Lemma 1. *If G_i is co-expressed under S_iS_j, S_iS_p and S_jS_p respectively, G_i must be co-expressed under $S_iS_jS_p$.*

Lemma 1 can guarantee using sample-growth method can generate bicluster in real-valued gene expression dataset without any information loss. Since our algorithm is based on sample-growth method, so the current extending differential co-expression bicluster under the same samples may have two parts of differential co-expression genes. Therefore, a differential co-expression bicluster can be denoted as $Samples(\boldsymbol{a}Genes, \boldsymbol{b}Genes)$ where $\boldsymbol{a}Genes$ is $Samples.PGenes$ and $\boldsymbol{b}Genes$ is $Samples.NGenes$. In above example, supposed the current extended sample set is S_1S_2, the current extended DC constant row bicluster is denoted as $S_1S_2(\boldsymbol{a}G_1G_2 - G_3 - G_4, \boldsymbol{b}G_3G_4G_5 - G_6)$. According to lemma 1, if one sample can be extended to the current extending DC bicluster when mining the real-valued microarray datasets, all the new generated edges should be satisfied the following definition.

Definition 5. *Supposed $S_i \cdots S_{j-1}S_j(\boldsymbol{a}G_x \cdots G_y, \boldsymbol{b}G_m \cdots G_n)$ be the current extending DC bicluster between two gene expression datasets and min_G is the minimum number of genes in the DC bicluster. If one sample S_p is a candidate sample, it should be satisfied as following: $|\ S_i \cdots S_{j-1}S_p.PGenes \cap S_i \cdots S_{j-1}S_j.PGenes \cap S_pS_j.PGenes\ | \geq min\_G$ or $|\ S_i \cdots S_{j-1}S_p.NGenes \cap S_i \cdots S_{j-1}S_j.NGenes \cap S_pS_j.NGenes\ | \geq min\_G$.*

Although above lemma and theorem guarantee our algorithm can use *Aprioir* property to produce DC constant row biclusters using sample-growth method, the mining procedure is very time-consuming. In order to increase the mining efficiency, our *DRCluster* algorithm generates maximal DC constant row biclusters without candidate maintenance. Traditional efficient maximal pattern mining technique is backward checking [5]. If there existed another extended priori candidate sample which can contain all the information of the current candidate sample, the current extended candidate sample would be pruned. However, above pruning technique can be used for mining maximal DC biclusters in discretized datasets, but it cannot be used to infer maximal DC constant row biclusters in real-valued datasets. For example, supposed the current extending sample is S_2, S_3 is the current extended candidate sample and S_1 is the priori candidate sample of S_2. Since $S_2S_3(\boldsymbol{a}G_1G_2 - G_3 - G_4, \boldsymbol{b}G_4 - G_6)$ is the subset of $S_1S_2(\boldsymbol{a}G_1G_2 - G3 - G4, \boldsymbol{b}G_4G_5 - G_6)$, so $S_2S_3(\boldsymbol{a}G_1G_2 - G_3 - G_4, \boldsymbol{b}G_4 - G_6)$ can be pruned when mining discretized datasets, but it cannot be used in real-valued microarray datasets. The reason is that G_3 is co-expressed under S_1 and S_2, S_2 and S_3, respectively. But it

is not co-expressed under S_1 and S_3. Therefore, although $S_2S_3(\boldsymbol{a}G_1G_2 - G_3 - G_4, \boldsymbol{b}G_4 - G_6)$ is the subset of $S_1S_2(\boldsymbol{a}G_1G_2 - G_3 - G_4, \boldsymbol{b}G_4G_5 - G_6)$, there could not generate a set of co-expressed genes under $S_1S_2S_3$, which is the superset of $S_2S_3(\boldsymbol{a}G_1G_2 - G_3 - G_4, \boldsymbol{b}G_4 - G_6)$. Therefore, the candidate sample S_3 of S_2 should not be pruned. If S_3 can be pruned, it must guarantee all the biclusters which are generated by extending $S_2S_3(\boldsymbol{a}G_1G_2 - G_3 - G_4, \boldsymbol{b}G_4 - G_6)$ are the subset of biclusters which are generated by extending $S_1S_2(\boldsymbol{a}G_1G_2 - G_3 - G_4, \boldsymbol{b}G_4G_5 - G_6)$. Based on above observation and analysis, the following lemma can guarantee mining maximal DC constant row biclusters without candidate maintenance in real-valued microarray datasets.

Lemma 2. *Given P be the current extending DC bicluster, M is the candidate sample set of P and N is the priori candidate sample set of P. Supposed the current candidate sample is $M_i(M_i \in M)$, and N_j is a priori candidate sample where $N_j \in N$. If $PM_i.PGenes$ should be pruned, it is satisfied the following criteria. (1) $PN_jM_i.PGenes$ is the same as $PM_i.PGenes$; (2) For each other candidate sample M_p in M, $PN_jM_i.PGenes$ is the subset of $PN_jM_p.PGenes$.*

Lemma 2 states how to escape of producing non-maximal DC biclusters without candidate maintenance. According to above lemma, *DRCluster* algorithm exploits the following pruning techniques to achieve mining maximal DC co-expression biclusters without candidate maintenance in real-valued microarray datasets.

Pruning 1. If the positive DC genes and the negative DC genes of the current extended candidate sample are both pruned based on Lemma 2, the current candidate sample would be pruned.

Pruning 2. If the positive DC genes of the current extended candidate sample is pruned based on Lemma 2 and the negative DC genes of the current extended candidate sample cannot be pruned, the positive DC genes of the current extended candidate sample would be pruned.

Pruning 3. If the negative DC genes of the current extended candidate sample is pruned based on Lemma 2 and the positive DC genes of the current extended candidate sample cannot be pruned, the negative DC genes of the current extended candidate sample would be pruned.

According to above lemmas and pruning techniques, *DRCluster* algorithm is designed for mining maximal DC biclusters without candidate maintenance in two real-valued gene expression datasets, which is shown in Algorithm 1. Fig.3 shows an example to illustrate the process of DRCluster for mining the examples datasets which are Table 1 and Table 2. The minimum number of genes and samples are both set to 2.

Input: Two real-valued microarray datasets: D_1 and D_2; the minimum number of genes in bicluster: min_G; the minimum number of samples in bicluster: min_S; WUR Graph: L; the current extending DC bicluster: P.
Output: the maximal DC bicluster set.
Initialization: $P = \emptyset$; $L = \emptyset$;
Method: `DRCluster` (D_1, D_2, min_G, min_S, L, P)
if $L = \emptyset$ **then**
  construct L;
end
scan L and find all the candidate set S of P;
foreach *candidate* $S_i \in S$ **do**
  if *the DC biclsuter does not satisfy* ***Pruning 1*** *and* ***Pruning 2*** *and* ***Pruning 3****, and the number of genes in PS_i is greater than min_G* **then**
    $P.Sample$ = $PS_i.Sample$;
    $P.PGenes$ = $P.PGenes \cap PS_i.PGenes$;
    $P.NGenes$ = $P.NGenes \cap PS_i.NGenes$;
    `DRCluster` (D_1, D_2, min_G, min_S, L, P);
  end
  else if *PS_i satisfies* ***Pruning 2*** **then**
    $P.Sample$ = $PS_i.Sample$;
    $P.PGenes$ = $P.PGenes \cap PS_i.PGenes$;
    $P.NGenes$ = Null;
  end
  else if *PS_i satisfies* ***Pruning 3*** **then**
    $P.Sample$ = $PS_i.Sample$;
    $P.NGenes$ = $P.NGenes \cap PS_i.NGenes$;
    $P.PGenes$ = Null;
  end
end
if *P is greater than any candidate bicluster of P and the number of samples in P is greater than min_S* **then**
  `Output` (P);
end
return;

Algorithm 1: DRCluster

4. EXPERIMENTAL RESULTS

In this section, we will present several experiments to evaluate the effectiveness and efficiency of our algorithm in real-valued microarray datasets. All approaches are implemented in Visual C++ and evaluated on an Intel(R) Core(TM)2 2.53GHz Duo CPU and 4G RAM running Windows 7. The performance of *DRCluster* algorithm will be compared with *SDC* [12] to produce maximal DC constant row biclusters in two real-valued microarray datasets. *SDC* algorithm uses SDC support to control the number of produce SDC biclusters. Low value of SDC support can produce more SDC biclusters and needs more time consuming. High value can produce outstanding SDC bicluster. Our implemented *SDC* bicluster algorithm only outputs the maximal subspace differential biclusters. Due to original *SDC* algorithm can only exploit SDC patterns in two gene expression datasets, the detail of our implemented *SDC* algorithm to mine DC constant row biclusters is described as following. (1) Since constant row bicluster aims to find the set of gene which has coherent expression values under the same set of experimental conditions, and *SDC* algorithm adopts a width-growth method to gener-

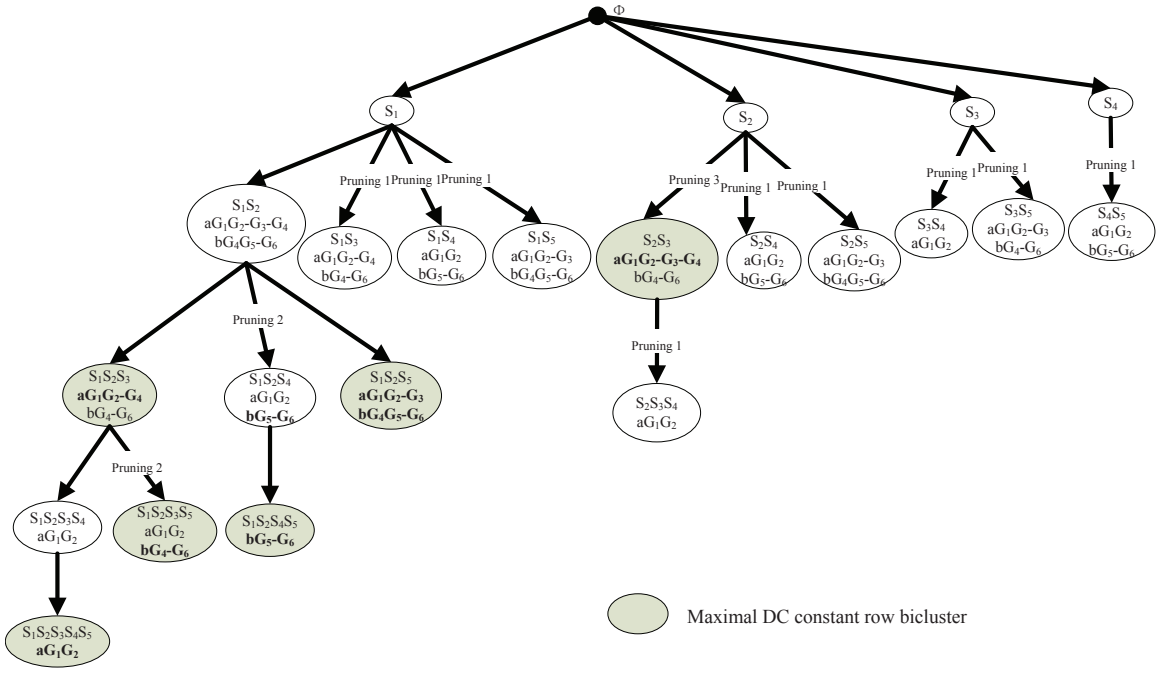

Figure 3 The process of DRCluster mining maximal DC constant row biclusters.

ate all the subspace differential patterns. Therefore, using *SDC* algorithm to produce constant row bicluster needs to infer all the coherent samples for each gene, which is the first step of our implemented *SDC* algorithm. (2) We use the differential *sample range support* to replace of differential *range support* [12] to mine subspace differential co-expression constant row biclusters. (3) Using the definition of subspace differential support shown in [12] to infer differential co-expression bicluster needs to be modified. Since the samples in bicluster may not be all the samples in the gene expression dataset. Therefore, we detect subspace differential co-expression biclusters whose genes show highly corrected co-expression under one set of samples in one dataset but not under the same set of samples in the other.

We used the real-valued gene expression datasets from *AGEMAP* [14], which is a large resource database that catalogs changes in gene expression as a function of age in mice. *AGEMAP* includes 8,932 genes and a number of 16,896 *cDNA* clones in 16 tissues as a function of age [14]. For each tissues, there are five male and five female mice aged 1, 6, 16, 24 month. In this paper, we will analyze three tissues, which are Hippocampus, Heart and Gonads respectively. Our purpose is to find potential co-expressed genes which are age-related. In this paper, we only use one male and one female mouse (denoted as *'c'* in *AGEMAP*) aged 6 month and 16 month to evaluate our algorithm. The number of conditions is 12.

A. Efficiency Comparison

In this section, we will compare the efficient performance of *DRCluster* algorithm with *SDC* algorithm. We applied them on different size of gene expression datasets to show their performance. The genes in each dataset are chosen by the order in *AGEAMP*. Since *SDC* algorithm uses *Apriori* concept to produce DC bicluster, so the efficiency is very low. As mentioned above, the larger of SDC support can result in less time consuming and better results. Therefore, we set the SDC support is 1, which can produce SDC bicluster efficiently and get outstanding results. Since our *DRCluster* produces DC biclusters without any differential co-expression support relaxation, so *DRCluster* is not set differential support. The minimum number of samples and genes in DC bicluster produced by above two algorithms are both set to 3. The minimum *sample range support* threshold is 2.

Fig.4 to Fig.6 show the runtime of each above algorithm with respect to various database sizes under various self-assignment values. When the database size increases, the runtime increases dramatically in each self-assignment value. And the larger self-assignment value results in the longer running time. The reason is that larger self-assignment value can get much greater scale of DC bicluster, which would influence the efficiency. It is shown *DRCluster* is more efficient than *SDC* for running time on different database sizes under different self-assignment values, except for when database has 100 and 200 genes under self-assignment value α be 0.3. Since larger self-assignment value may result in less overlap percent and larger scale of DC bicluster. So when the database is smaller,

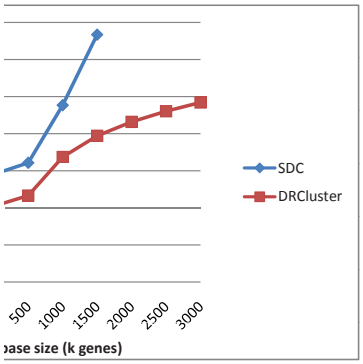

Figure 4 The runtime of each algorithm on different size database at α=0.1.

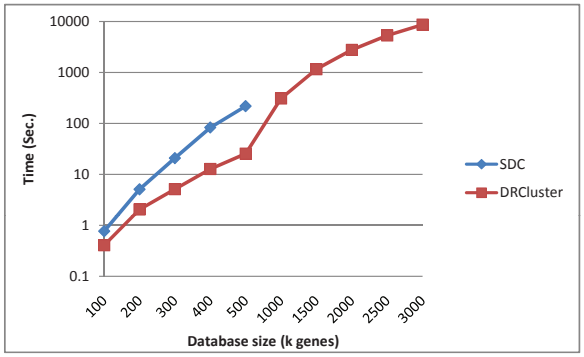

Figure 5 The runtime of each algorithm on different size database at α=0.2.

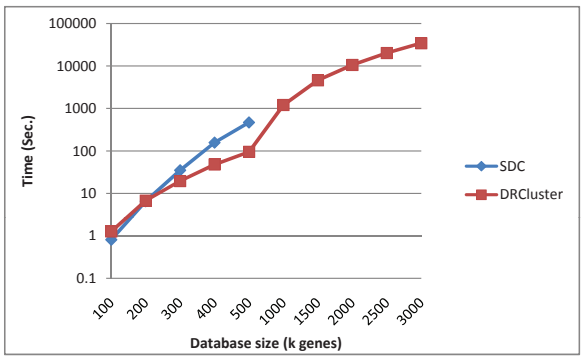

Figure 6 The runtime of each algorithm on different size database at α=0.3.

most of produced biclusters may be maximal. Our designed pruning technique of *DRCluster* needs to check more times for pruning but pruned less, which may influence a little efficiency (however, the difference gap is within one second).

Due to the efficient pruning technique of *DRCluster*, it can produce all the maximal DC constant row biclusters within 90 seconds when the database size is not greater than 1500 when α is 0.1, which is shown in Fig.4. On 1500 genes, *DRCluster*(87.71s) is almost 534 times faster than *SDC*(46893.17s). Due to the less efficiency of *Apriori-like* concept in *SDC* bicluster algorithm, it cannot terminate when mining 2000 genes or more. When the database is increased to 3000 genes, *DRCluster* can terminate within 700 seconds. As discussed in above, larger self-assignment value α may produce greater scale of biclusters. As shown in Fig.5 and Fig.6 SDC algorithm cannot terminate when the number of genes is greater 1000 when α is 0.2 or 0.3. However, *DRCluster* can produce all the maximal DC constant row biclusters without candidate maintenance in memory, so it can terminate on mining 3000 genes datasets in Fig.5 and Fig.6.

B. Significance of DC Biclusters Using MSE Tests

In this section, we measured the coherence of each SDC patterns using the MSE score. The *mean squared error* (MSE) has been proposed by [1] to measure the coherence of expression levels of a subset of genes across a subset of experimental conditions. MSE can be used to capture the coherence of expression levels of a subset of genes across a subset of experimental conditions. If I and J are the set of genes and samples in one bicluster, and D_{ij} is the expression value of ith gene under jth sample, The MSE score is defined as $M(I,J) = \frac{1}{|I||J|} \sum_{i\in I, j\in J} (D_{ij} - D_{iJ} - D_{Ij} + D_{IJ})^2$, where $D_{iJ} = \frac{1}{|J|} \sum_{j\in J} D_{ij}$ and $D_{Ij} = \frac{1}{|I|} \sum_{i\in I} D_{ij}$ are the means of the values in the ith row and jth column respectively, while $D_{IJ} = \frac{1}{|I||J|} \sum_{i\in I, j\in J} D_{ij}$ is the overall mean of the bicluster. The minimum value of MSE is obtained when the bicluster is constant row or column bicluster. MSE can also be used to measure the differential co-expression constant row bicluster. If the coherence of expression levels of all the genes of one DC bicluster under subset of experimental conditions in one microarray dataset is very higher (lower MSE score) and lower (higher MSE score) in the other microarray dataset, we claim this DC bicluster is significant.

We will test the potential age-related biclusters which are differential co-expression between 6 month and 16 month in the following paper. Since all the cDNA clones in *AGEMAP* cannot be potential age-related, [13] collected a list of 305 cDNA clones that are age-related in multiple mouse tissues. In the following experiments, we analyze the differential co-expression biclusters discovered by above two algorithms on these 305 cDNA clones. All the minimum *sample range support* of the following evaluated DC constant row bicluster is 1.5 and the differential support of *SDC* is 1 which can mine the most outstanding results.

Due to the limitation of paper space, we only show the distribution of MSE scores of DC constant row biclusters at self-assignment value $\alpha = 0.1$. The minimum number

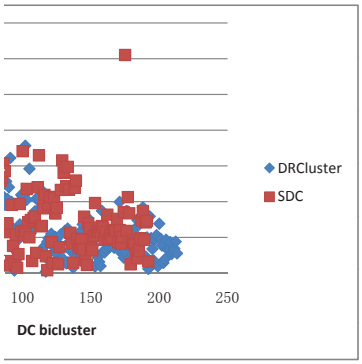

Figure 7 The MSE scores distribution of DC constant row biclusters produced from 6 month to 16 month in 6 month microarray.

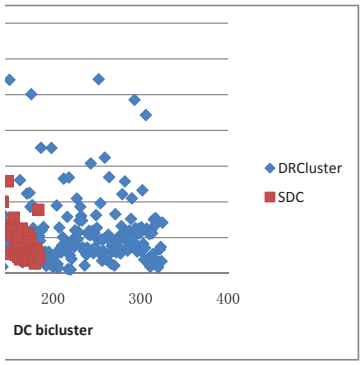

Figure 9 The MSE scores distribution of DC constant row biclusters produced from 16 month to 6 month in 6 month microarray.

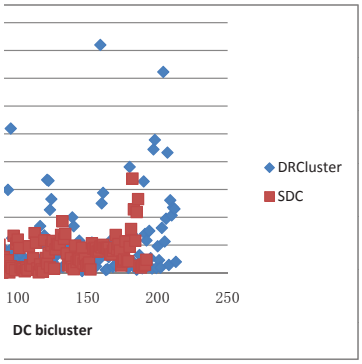

Figure 8 The MSE scores distribution of DC constant row biclusters produced from 6 month to 16 month in 16 month microarray.

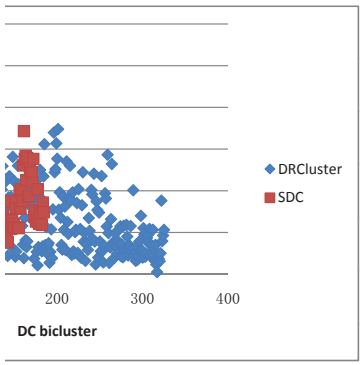

Figure 10 The MSE scores distribution of DC constant row biclusters produced from 16 month to 6 month in 16 month microarray.

of genes and samples are both 3. The MSE distribution of DC constant row biclusters that were produced from 6 month to 16 month in Fig.7 is evaluated in 6 month microarray. It means that the genes in bicluster are co-expressed in 6 month and not co-expressed or opposite co-expressed in 16 month. If the MSE scores are more lower-valued, the evaluated biclusters are better significance. It can be clearly seen that *DRCluster* can produce more and lower MSE-valued DC constant row biclusters than SDC algorithm in 6 month microarray dataset. Fig.8 shows the MSE score distribution of DC constant row biclusters in 16 month microarray. If the values are higher, it can be seen as significance. It can be seen from Fig.8, *DRCluster* can produce more totals and lower MSE-valued DC constant row biclusters than *SDC* algorithmin in 16 month microarray dataset. Fig.9 and Fig.10 show the MSE distributions of DC constant row biclusters that were produced from 16 month to 6 month are evaluated in 6 month gene expression data and 16 month gene expression data. Since such biclusters were inferred from 16 month to 6 month, so the distribution is opposite to Fig.7 and Fig.8. If the MSE score of bicluter is high in 6 month gene expression data and low in 16 month, such bicluster shows the significant coherent differential co-expression. Fig.9 to Fig.10 shows *DRCluster* can find more totals and lower MSE-valued DC constant row biclusters than *SDC* algorithm in 16 month gene expression data, more and high MSE-valued in 6 month. Therefore, based on Fig.7 and Fig.10, it shows that our produced DC constant row biclusters are more coherent than *SDC* algorithm's at $\alpha = 0.1$.

C. Biological Analysis

We will show how the biological significance of the DC biclusters found by each algorithm is evaluated in this section. We assess the DC bicluster quality by determining the percentage of functionally homogeneous DC biclusters among all identified DC biclusters. We used the Gene Ontology (GO) [15] annotation to test our results. If the ratio of one DC biclusters genes having the same known annotations which belong to an annotated GO functional category is greater than the user-defined threshold, this DC bicluster

is claimed as biological interesting one [16]. In this section, we analyze the differential co-expression biclusters discovered by each algorithm on 305 potential age-related cDNA clones. All the minimum *sample range support* of the evaluated DC constant row bicluster is 2, the minimum number of genes in bicluter is 4 and the differential support of *SDC* is 1.

Fig.11 to Fig.13 show the number of DC biclusters which are produced by each algorithm, evaluated by GO at different GO homogeneous threshold under different self-assignment value. We found that the number of enriched GO categories in each algorithm decrease with homogeneous threshold. It can also be clearly seen from these figures that, *DRCluster* can produce more GO-evaluated DC constant row biclusters than *SDC* algorithm at each homogeneous threshold under each self-assignment value. We found that our *DRCluster* can find the number of GO-evaluated DC constant row biclusters gap decreases with self-assignment value. *DRCluster* can find almost 18 times more than *SDC* when homogeneous threshold is 0.5 and self-assignment value is 0.1.

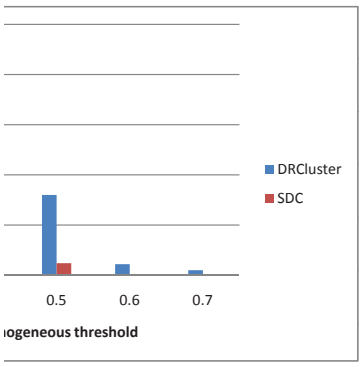

Figure 11 The number of GO-evaluated DC constant row biclusters comparison at α=0.1.

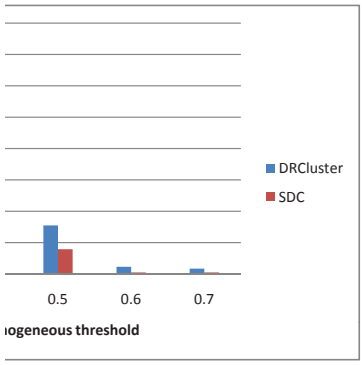

Figure 12 The number of GO-evaluated DC constant row biclusters comparison at α=0.2.

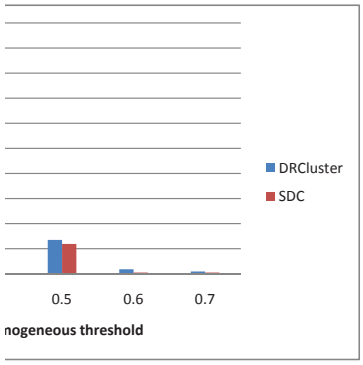

Figure 13 The number of GO-evaluated DC constant row biclusters comparison at α=0.3.

5. Conclusion

In this paper, we propose an algorithm, *DRCluster*, to mine maximal differential co-expression constant row biclusters in two real-valued gene expression datasets efficiently. *DRCluster* can find maximal DC constant row biclusters without candidate biclusters maintenance in memory. Compared with the existing *SDC* bicluster mining algorithm, it is shown that our algorithm is more efficient. The experiments show our algorithm can produce more biological significance. However, there remain several further investigations. We discuss some limitations of *DRCluster* algorithm and our future work. (1) *DRCluster* mines DC constant row biclusters in two gene expression datasets. Mining differential co-expression biclusters in several microarray datasets may be more interesting biologically. In the future, we plan to infer DC biclusters in several real-valued gene expression datasets. (2) Our *DRCluster* cannot handle any fault-tolerant biclusters. Gene expression dataset has lots of noisy value. Biologically speaking, fault-tolerant capability [17,18] is a key meteyard for bicluster algorithms. Next, we also plan to extend our approach to infer differential co-expression fault-tolerant biclusters in real-valued gene expression datasets.

Acknowledgement

The work is supported by the National Key Basic Research Program of China under Grant No.2012CB316203. It is also partly supported by the National Natural Science Foundation of China under Grant No.61033007 and No.60970065, the Research Foundation at Northwestern Polytechnical University of China under Grant No.JC201042. This material is also based upon work funded by Zhejiang Provincial Natural Science Foundation of China under Grant No.R1110261.

References

[1] Y. Cheng and G.M. Church. Biclustering of Expression Data. Proc. 8th Intl Conf. Intelligent Systems for Molecular Biology, (2000) ACM Press, 2000, pp. 93C103.

[2] A. Subramanian, P. Tamayo, V. Mootha, S. Mukherjee, B. Ebert, M. Gillette, A. Paulovich, S. Pomeroy, T. Golub and E. Lander. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. PNAS, 102(43):15545C15550 (2005).

[3] A. Ben-Dor, B, Chor, R, Karp and Z. Yakhini. Discovering local structure in gene expression data: the order-preserving submatrix problem. J Comput Biol, 10:373-384 (2003).

[4] L. Zhao, and M. Zaki. MicroCluster: Efficient deterministic biclustering of Microarray data. IEEE Intelligent Systems, 20(6):40-49 (2005).

[5] M. Wang, X.Q. Shang, S.H. Zhang and Z.H. Li. FDCluster Mining frequent closed discriminative bicluster without candidate maintenance in multiple microarray datasets. (2010) Proceedings of ICDM Workshops 2010: 779-786.

[6] D. Kostka, and R. Spang. Finding disease specific alterations in the coexpression of genes. Bioinformatics, 20 (Suppl. 1): i194-i199 (2004).

[7] Y. Okada and T. Inoue. Identification of differentially expressed gene modules between two-class DNA microarray data. Bioinformation, vol. 4, no. 4, pp. 134C137 (2009).

[8] A. Serin and M. Vingron. Debi: Discovering differentially expressed biclusters using a frequent itemset approach. Algorithms for Molecular Biology, vol. 6, no. 1, p. 18 (2011).

[9] D. Burdick, M. Calimlim, and J. Gehrke. Mafia: A maximal frequent itemset algorithm for transactional databases. (2001) In proceedings of ICDE 2001, pp. 443C452.

[10] Lucinda K. Southworth, Art B. Owen, Stuart K. Kim. Aging Mice Show a Decreasing Correlation of Gene Expression within Genetic Modules. PLoS Genet 5(12): e1000776 (2009).

[11] O. Odibat, C. K. Reddy and C. N. Giroux. Differential biclustering for gene expression analysis. (2010) In Proceedings of the ACM Conference on Bioinformatics and Computational Biology (BCB),pages 275C284. ACM.

[12] G. Fang, R. Kuang, G. Pandey, M. Steinbach, Chad L. Myers and V. Kumar. Subspace Differential Coexpression Analysis: Problem Definition and A General Approach. (2010) Proceedings of the 15th Pacific Symposium on Biocomputing(PSB), 15:145-156.

[13] G. Pandey, G. Atluri, M. Steinbach, C. L. Myers and V. Kumar. An association analysis approach to biclusting. (2009) In Proc. ACM Conf. on Kownlege Discovery and Data Mining, pages 677-686.

[14] J.M. Zahn, S. Poosala. AGEMAP: A gene expression database for aging in mice. PLOS Genetics, 3(11):2326-2337 (2007).

[15] The Gene Ontology Consortium. The gene ontology (go) database and informatics resource. Nucleic Acids Research 32, 258C261 (2004).

[16] M. Wang, X.Q. Shang, D. Xie and Z.H. Li. Mining frequent dense subgraphs based on extending vertices from unbalanced PPI networks. (2009) In proceedings of the 3rd International Conference on Bioinformatics and Biomedical Engineering. Beijing, China: IEEE pp 978-1-4244-2902-8.

[17] M. Wang, X.Q. Shang, M. Miao, Z.H. Li and W.B. Liu. MFCluster Mining maximal fault-tolerant constant row biclusters in microarray dataset. (2011) Proceeding of WAIM 2011, p 181-190.

[18] M. Wang, X.Q. Shang, M. Miao, Z.H. Li and W.B. Liu. FT-Cluster: Efficient Mining Fault-Tolerant Biclusters in Microarray Dataset. Proceeding of ICDM 2011 workshops, p 1075-1082.

Miao Wang is a doctoral student at The School of Computer Science and Engineering at the Northwestern Polytechnical University, Xi'an, China. He completed his Master Degree from University of Northwestern Polytechnique in 2008. Since 2006 he has been researching in data mining and bioinformatics. He is a student member of China Computer Federation.

Xuequn Shang is a professor at The School of Computer Science and Engineering at the Northwestern Polytechnical University, Xi'an, China. She completed her PhD degree from University of Magdeburg, Germany in 2005. She is head of computer software and engineering department. She is a member of China Computer Federation. Since 2001, she has been researching in data mining, database technology and bioinformatics. Prof. Shang has been PI for one NSF grant.

Xiaoyuan Li is a graduated student at The School of Computer Science and Engineering at the Northwestern Polytechnical University, Xian, China. He completed his Bachelors Degree at College of Information Engineering at North China University of Water Resources and Electric Power in 2010. He has been researching in data mining and bioinformatics since 2010.

Zhanhuai Li is a professor at The School of Computer Science and Engineering at the Northwestern Polytechnical University, XiAn, China. He is a senior member of China Computer Federation. He is vice chairman of Database Technical Committee, China Computer Federation. He received his Masters and Ph.D. degrees from Northwestern Polytechnical University in 1987 and 1996 respectively. His research interest is database technology, software engineering and data mining. Prof. has been PI for grants, including NSF, 863 and 973.

Wenbin Liu is a professor at school of Physics and Electronic information engineering in Wenzhou University. Research interests mainly include DNA computing, computational biology, pattern recognition and data mining. Form 2001 to 2004, study for PhD in Huazhong University of science and technology. From 2007 to 2008, visit Ilyas group at the Institute for Systems Biology in Seattle. Up to now, Pro. Liu has been PI for two NSF grants and two ZJNSF grants.