BiCross : A Biclustering Technique for Gene Expression Data using One Layer Fixed Weighted Bipartite Graph Crossing Minimization

Suvendu Kanungo
Dept. of Comp. Sc.
BIT Mesra, Ranchi
Allahabad Campus, UP, India

Gadadhar Sahoo
Dept. of IT
BIT Mesra, Ranchi
Jharkhand, India

Manoj Madhava Gore
Dept. of Comp. Sc. & Engg.
MNNIT, Allahabad
UP, India

ABSTRACT

Biclustering has become an important data mining technique for microarray gene expression analysis and profiling, as it provides a local view of the hidden relationships in data, unlike a global view provided by conventional clustering techniques. This technique, in contrast to the conventional clustering techniques, helps in identifying a subset of the genes and a subset of the experimental conditions that together exhibit co-related pattern. In this paper, a biclustering technique using weighted crossing minimization paradigm is proposed, which can mine significant patterns by employing a local search instead of a global search of the input data matrix. We present the novel idea of modelling the gene expression data as a weighted bipartite graph between genes and experimental conditions in order to rearrange the vertices in one layer of this graph. Using this model, an efficient biclustering technique is developed that can mine different types of biclusters and works well in practice for simulated and real world data. The experimental results demonstrate that, our method is scalable to practical gene expression data and has superiority over other similar algorithms in terms of accuracy and computational efficiency.

General Terms

Data Mining, Clustering, Algorithms

Keywords

Crossing minimization, Biclustering, Gene Expression Data, Bipartite Graph.

1. INTRODUCTION

Microarray technology makes it possible to simultaneously study the expression of thousands of genes during a single experiment. Data arising from microarray experiment which is called gene expression data are usually arranged in a co-occurance table or matrix, such as a gene-condition table, where rows represent genes and columns represent experimental conditions or samples. Each element of this matrix represents the expression level of a gene under a specific condition, and is represented by a real number, which is usually the logarithm of relative abundance of mRNA of the gene under specific condition. Recently, biclustering has created great interest in research community due to the challenges associated with high dimensionality of data sets. Many conventional clustering algorithms [29] have been developed to mine clusters in the whole data space. Unfortunately, most of these conventional clustering algorithms do not scale well to cluster high dimensional data sets in terms of effectiveness and efficiency because of their inherent sparsity. This difficulty motivates the concept of biclustering, co-clustering or subspace clustering. Biclustering has become an important technique to microarray gene expression data analysis, as it provides a local view of the hidden relationships in data, unlike a global view provided by traditional clustering techniques. This technique has important applications in many fields, for example, e-commerce, data mining, pattern recognition, statistics, machine learning and computational biology.

The complexity of biclustering problem largely depends on the problem formulation. Our approach is based on crossing minimization of a weighted bipartite graph. Minimum linear arrangement problem (MinLA) or optimal linear ordering is commonly employed in VLSI circuit layout in order to minimize the total wire length. MinLA is also connected with crossing minimization in the context of bipartite graph drawing [13]. It has been shown by Pach et al. [12] that for a large class of bipartite graphs, reducing the bipartite crossing number is equivalent to reducing the total edge length. This technique deals with proper placement of similar or related nodes in a graph, which motivates us for employing crossing minimization technique to our problem formulation. Basically there are two variants in bipartite crossing minimization technique discussed in literature in the context of graph drawing, namely two layer free crossing minimization (TLFCM) and one layer free crossing minimization (OLFCM). In OLFCM, the ordering of one layer is already fixed. In order to solve TLFCM problem, OLFCM solution is commonly applied iteratively by alternating the fixed layer at each iteration. It has been shown that optimal crossing minimization is NP-hard [9] and as a result extensive research has been devoted to the design of heuristics and approximation algorithms for these problems with reasonable accuracy.

In this paper our contributions are:-

- We model the gene expression data as a weighted bipartite graph between genes and experimental conditions.
- For conditioning (outlier removal) of input data, we devise a method in which each normalized sample data is partitioned into unequal length intervals based on mean values.
- We employ an approximation algorithm for one layer weighted crossing minimization of bipartite graph with $O(|E| + |V_0| + |V_1|\log|V_1|$, where vertex set

$V = V_0 \cup V_1$ and E is the set of edges.

- We provide an efficient local search based algorithm for bicluster identification from conditioned data using KL distance [28], which can extract constant, coherent and overlapped biclusters in the presence of noise.

2. RELATED WORK

Several approaches have been proposed for solving biclustering problem. As it is known to be NP-hard problem, several algorithms for mining biclusters use heuristics method or probabilistic approximation. An illustrative discussion on many of these algorithms can be found in [8][22].

Cheng and Church [1] identify biclusters with the help of mean squared residue score, which is a measure of the coherence of rows and columns in the bicluster. Here the user has to input a value of mean residue score δ and the number of biclusters to be extracted. This method involve several iterations and each iteration mine only a single bicluster while previously identified biclusters are masked with random values. However they did not address the issue of noisy data, where as in this paper we concentrate on noisy data.

Tanay et al. [2] introduced SAMBA, in which the data are modelled as a bipartite graph with genes corresponding to vertices in one bipartition and samples corresponding to vertices in other bipartition, where edges representing significant changes in expression. Edges and non-edges are weighted by likelihood scores derived from a probabilistic model for the bipartite graph. A bicluster is defined as a heavy subgraph, where the weight of the subgraph is the sum of the weights of the corresponding edges and non-edges. It repeatedly finds the maximal highly-connected subgraph in the bipartite graph and perform local improvement by adding or deleting a single vertex until no further improvement is possible. In order to avoid exponential runtime, they assumed that row vertices have d-bounded degree. However, our technique can handle graphs of arbitrary degrees.

Ben-Dor et al. [3] proposed OSPM, in which a bicluster is defined as a cluster of genes with the same rank profile across the biclustered samples. This method can mine large and statistically significant biclusters with the help of a greedy algorithm for identifying a fixed pattern of rows in a data set, one at a time. The time complexity of this technique is $O(nm^2l)$, where n and m are the number of rows and columns of the data matrix and l is the number of biclusters, which is slower than our approach.

Ahsan and Amir[4] identify biclusters by recursively removing noise with the help of crossing minimization technique. This method is based on binary representation of the bipartite graph corresponding to input data matrix. It is difficult to mine coherent biclusters, as this method use a static discretization of the input data matrix.

Wang et al. [5] proposed RMSBE, which can identify optimal square biclusters with the maximum similarity score. This method performs multiple scans of the data matrix in order to compute similarity score, reference gene identification and bicluster identification. The time complexity of this technique is $O(nm(n+m)^2)$, where n is number of rows and m is number of columns. Due to this cubic nature of complexity, it is not feasible for very high dimensional data. Prelic et al. [6] proposed BiMax, which can identify constant biclusters. This method discretize the input expression matrix into a binary matrix based on a threshold value. Therefore it is difficult to identify coherent biclusters.

Dhillon et al. [7] proposed a Bregman Divergence based loss function, which is applicable to all density functions belonging to exponential family, for identifying clusters. Our approach employs KL distance to measure similarity between two rows to identify constant biclusters.

Bergmann et al. [23] proposed the iterative signature algorithm (ISA) that uses gene signatures and condition signatures in order to extract biclusters with both up and down-regulated expression values. They identify several transcription modules (biclusters) by executing the algorithm on reference gene sets. The reference gene sets needs to be carefully selected for extraction of good quality biclusters.

3. BASIC CONCEPTS AND MODEL FORMULATION

The purpose of this section is to present basic concepts, definitions and model formulation. Let $G=\{g_0,g_1,...,g_{n-1}\}$ be a set of n genes and $S=\{s_0,s_1,...s_{m-1}\}$ be a set of m biological samples. A $2-D$ microarray data-set is a real valued $n \times m$ expression matrix $D=G\times S=\{d_{ij}\}$ where $i\in[0,n-1]$, $j\in[0,m-1]$ and each entry d_{ij} corresponds to the logarithm of the relative abundance of mRNA of a gene under a specific sample S_j. A bicluster corresponds to a sub matrix that exhibits some coherent tendency. Let B be a sub matrix of dataset D. Bicluster $B=X\times Y=\{b_{ij}\}$ where $X\subseteq G$ and $Y\subseteq S$, provided certain conditions of homogeneity are satisfied. We define the volume or size of a bicluster B as the number of elements d_{ij}, such that $i\in X$ and $j\in Y$.

Let $B_{2,2}=\begin{bmatrix} b_{ip} & b_{iq} \\ b_{jp} & b_{jq} \end{bmatrix}$ be any arbitrary sub matrix of B.

Then B is a scaling bicluster if $b_{iq}=\alpha_i b_{ip}$ and $b_{jq}=\alpha_j b_{jp}$; and $\left|\alpha_i-\alpha_j\right|\le\rho$, where α is a constant multiplicative factor. B is a shifting bicluster iff $b_{iq}=\beta_i+b_{ip}$ and $b_{jq}=\beta_j+b_{jp}$; and $\left|\beta_i-\beta j\right|\le\rho$, where β is constant additive factor. B is a constant bicluster if $b_{ij}=\mu$ or $b_{ij}\approx\mu$. B is a constant row bicluster if $b_{ij} = \mu + \alpha_i$ or $b_{ij} = \mu \times \alpha_i$. Similarly B is a constant column bicluster if $b_{ij} = \mu + \beta_j$ or $b_{ij} = \mu \times \beta_j$, where μ is a typical value within the bicluster; α_i and β_j are adjustment for row and column respectively. B is overlap bicluster if b_{ij} is the sum or product of the contribution of different biclusters to which they belong.

Definition 3.1 Bipartite Graph: A graph $G(V,E)$ is called Bipartite if its vertex set V can be decomposed into two disjoint subsets V_0 and V_1 (i.e. $V = V_0 \cup V_1$) such that every edge in E joins a vertex in V_0 with a vertex in V_1 (i.e. $V_0 \cap V_1 = \phi$).

Definition 3.2 Weighted Bipartite Graph: A graph $G(V_0,V_1,E,W)$ is called weighted bipartite graph if $W = (w_{ij})$ where $w_{ij} \geq 0$ denotes the weight of the edge $\{i,j\}$ between vertices i and j .

Definition 3.3 Bipartite Drawing: A bipartite drawing or 2-layer drawing of $G(V_0,V_1,E)$ is a graph representation where the nodes of V_0 and V_1 are placed in two parallel lines $y = 0$ and $y = 1$, while the edges are drawn with straight lines between them.

Definition 3.4 Crossing Number: Let h be a bipartite drawing of bipartite graph $G(V_0,V_1,E)$. Let $bcr_h(e)$ represent the number of crossing of the edge $e \in E$ with other edges of E . Let $bcr(h)$ represent the total number of crossings in h i.e. $bcr(h) = \frac{1}{2}\sum_e bcr_h(e)$. The bipartite crossing number of G denoted by $bcr(G)$ is the minimum number of crossings over all bipartite drawings of G i.e. $bcr(G) = \min_h bcr(h)$.

Definition 3.5 Weighted Crossing: Let two edges $e_1, e_2 \in E$ of a bipartite drawing cross each other with nonnegative weights $w(e_1)$ and $w(e_2)$ respectively. Then, this crossing amount to $w(e_1) \times w(e_2)$ in the total weighted crossings.

Most popular heuristics for crossing minimization include the barycenter method [10] and the median heuristics [11]. A survey of various heuristics has been taken up in detail in [14] and shown that the barycenter method produces better results than the median heuristics. Let us illustrate the benefit of the commonly used barycenter heuristic [10], where we order vertices of a layer according to means of their adjacent vertices in other layer. By repeating this ordering process in turns in two layers until an equilibrium state is reached, we get ordering of vertices which minimize the number of edge crossings. Each iteration of the barycenter heuristic can be implemented in $O(|E| + |V|\log|V|$ time. Figure 1 shows how the adjacency matrix of the corresponding bipartite graph changes when applying the barycenter heuristic.

After crossing minimization, similar vertices are brought together in order to form biclusters. For example Figure 1(c) shows reordered bipartite graph and the corresponding adjacency matrix of Figure 1(d) shows two biclusters $B_1 = \{B,D\} \times \{P,R,Q\}$ and $B_2 = \{A,C\} \times \{Q,S,T\}$.

4. THE PROPOSED TECHNIQUE

For ease of understanding, our technique consists of the following basic steps:

a. Normalization and outlier removal from data in D
b. Weighted crossing minimization of G to get G'
c. Extraction of biclusters from reordered bipartite graph G'
d. Evaluation of extracted biclusters B

4.1 Normalization and Outlier Removal

Gene expression data is usually noisy and may contain missing values. Prediction of missing value has been taken up in great detail in [15] and [16]. Therefore proper conditioning of data is essential before applying the clustering algorithm. In order to handle missing values, we have adopted the approach used in [17] i.e. replacing all missing values by zero. We normalize each sample of data with a mean of 0 and variance of 1. In order to handle outlier efficiently, we have partitioned the normalized sample data into unequal length intervals based on mean value. The motivation of considering unequal length intervals is due to the ineffectiveness of equal length intervals for extreme outlier values. Also the decision of number of intervals may lead to inappropriate interval boundaries, as it does not depend on the properties of data [18]. In this approach, we partition the sample data into two halves with the mean value. Then, recursively each half is partitioned again into two halves with its own mean. This process proceeds until each sample has been partitioned into required number of intervals. The number of intervals (r) for each sample depends on the data size (n). Further, to have balanced partition, we assume $r = 2^k$, where k is a positive integer and $r^2 \times 35 \leq n$, where 35 is the minimum sample size for large sample procedures [19]. The values within the intervals are then smoothed by interval means. We found that this way of partitioning is very effective and can deal with outliers efficiently. We create a probability distribution table during this process in order to compute similarity between two rows based on KL distance.

4.2 Weighted Crossing Minimization

Our approach employ weighted version of OLFCM, called WOLFCM, which is NP-complete in nature even when the graph instances are sparse [20], with a fixed ordering for sample vertices in layer V_0 and dynamic gene vertices in layer V_1, and nonnegative weight for edges. The motivation of this approach is to cater for practical non-binary gene expression data matrices and identification of coherent biclusters with better time complexity. The nodes in V_0 are labeled from 1 to m. For every node in V_1, we store its adjacent nodes label in a set. We partition V_1 into $|V_0|$ partitions and decide the proper partition for a node in V_1 based on its left weighted sum and right weighted sum. Starting from the first adjacent node of any node in V_1, we increase the left weighted sum value each time and decrease the right weighted sum value whenever necessary. We find the partition for a node in V_1 when the left weighted sum value is greater than or equal to the right weighted sum value. Then based on the amount of weighted crossing between any two nodes in V_1, we decide the proper place a node in the corresponding partition. Pseudocode for one layer free weighted crossing minimization of a bipartite graph is given in Algorithm-I. By applying Algorithm- I to the bipartite graph in Figure 1, we also get similar biclusters i.e. $B_1 = \{B, D\} \times \{P, R, Q\}$ and $B_2 = \{A, C\} \times \{Q, S, T\}$. After weighted crossing minimization, similar vertices of layer V_1 are brought closer enabling us a local search in order to extract different types of biclusters.

4.3 Bicluster Extraction

After reordering rows using weighted crossing minimization process, similar rows are placed close to each other which enable us for a local search in order to extract different types of bicluster. This process use the probability distribution table built during data preprocessing. Pseudocode for bicluster extraction is given in Algorithm - II. The extraction process starts from the first element of reordered matrix and use two different distance measures, KL distance (relative entropy) and Bscore, in order to check for similarity between rows and columns respectively. Our approach is row major, as it compare two rows simultaneously for coherency. KL distance for two probability mass functions p(x) and q(x) can be defined as:

$$KL(p \,\|\, q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} \qquad (1)$$

We have employed KL distance between two consecutive rows for a set of columns in order to extract constant biclusters. This distance is computed using the probability distribution table. Two adjacent columns can be compared for coherence by computing its score, called Bscore, which can be defined as:

$$Bscore = \left|(b_{i_1 j_1} + b_{i_2 j_2}) - (b_{i_1 j_2} + b_{i_2 j_1})\right|, \qquad (2)$$

where i_1 and i_2 are rows of a bicluster and j_1 and j_2 are columns of a bicluster; and b_{ij} is the value in a bicluster. This technique is based on additive model and hence can extract additive coherent biclusters. Bscore is computed using the reordered matrix and can extract coherent clusters. Initially, we keep a reference set of matching columns based on the threshold values of KL and Bscore. Columns are added to this set if KL < δ or Bscore < β. If subsequent rows contain matching columns, they are added to the row cluster. For overlap condition, if subsequent rows contain more matching columns then the overlap flag is set. In this case, we preserve the row and column location in order to start locating biclusters from this row and column in the next iteration of the extraction process. When we reach end of the data matrix and current row and column cluster contain required number of rows and columns, then we store this bicluster in the bicluster set. If the current column cluster is empty, then we declare the current row cluster and reference column cluster as a bicluster provided that they satisfy the minimum row and column constraint.

4.4 Complexity Analysis

In the first stage for Algorithm-I, normalized and conditioned gene expression data is reordered by portioning layer V_1 nodes (rows) into $|V_0|$ (columns) partitions. The correct partition for a node in V_1, which occur when L_{wsum} value is larger than R_{wsum}, would take time $O(|E| + m + n)$. Then placing this node in correct position in the corresponding partition would take time $O(m + n \log n)$. Therefore, the total time for reordering the rows of D would take $O(|E| + m + n \log n)$. In the second stage for Algorithm-II, it would take time $O(|B| R_B C_B)$ for disjoint biclusters, where as in case of overlapping biclusters it would take time $O(o_d |B| R_B C_B)$, where R_B and R_C denote average number of rows and columns in a bicluster respectively; and o_d denote the degree of overlap among biclusters.

4.5 Implementation

We have implemented our proposed algorithm in C++ under windows environment on a computer with configuration of Core 2 Duo 2.2 GHz of CPU and 3 GB RAM. We evaluate its accuracy and performance using synthetically generated dataset and real dataset. For real gene expression dataset, we have considered the model organism Saccharomyces Cerevisiae, provided by Gasch et al. [21], since the yeast GO annotations are more extensive compared to other organisms. This gene expression dataset contains 2,993 genes and 173 different stress conditions.

5. EXPERIMENTAL RESULTS AND COMPARISION WITH OTHER ALGORITHMS

It is very difficult to have a comparative study among various biclustering algorithms in view of different problem formulation with different clustering criterion. As a result a biclustering algorithm works well in certain circumstances and perform poorly in others. Therefore, we need to define a common setting for all these algorithms in order to have a fair comparative study. Our main objective is to validate our proposed algorithm for identification of constant, coherent and overlapped biclusters amid noise while comparing with other biclustering algorithms. Also, we illustrate the biological significance of extracted biclusters in gene expression data. For comparison purpose, we consider other similar algorithms like CC [1], ISA [23], OSPM [3], SAMBA [2] and BiMax [6]. In order to evaluate, we used the Bicluster Analysis Toolbox (BicAT) developed by Prelic et al. [25] for implementation of BiMax, CC, ISA and OSPM. Also, we have used EXPANDER developed by Maron-Katz et al. [24] for implementation of SAMBA.

5.1 Synthetic Dataset

In case of synthetic gene expression data, we used the technique proposed by Zimmermann et al. [6] for evaluation of implanted constant, coherent and overlap biclusters. For constant bicluster generation, we adopt the following steps:

a. Generate a 100 × 100 matrix A with all elements 0

b. Generate ten biclusters (modules) of size 10 × 10 with all elements 1

c. Replace elements of biclusters with random noise values from uniform distribution (-σ,σ)

d. Implant the ten biclusters into A without overlap

For all experimentation, we set the noise level range from 0.0 to 0.25. In case of overlapping biclusters, we used 10 degrees of overlap (o_d = 0,1,2,3,4,5,6,7,8,9) , where the size of matrix and bicluster vary from 100 × 100 to 110 × 110 and from 10 × 10 to 20 × 20, respectively. The steps for evaluation of coherent biclusters are same as that of constant bicluster, but rows and columns in a bicluster have a 0.02 increasing trend. In order to validate the accuracies of different algorithms, we apply the gene match score proposed by Zimmermann et al. [6]. Let M_1 and M_2 be two sets of biclusters. The match score of M_1 with respect to M_2 is given by:

$$S_G(M_1,M_2)=\frac{1}{|M_1|}\sum_{(G_1,S_1)\in M_1}\max_{(G_2,S_2)\in M_2}\frac{|G_1\cap G_2|}{|G_1\cup G_2|}, \qquad (3)$$

where G and S are set of genes and a set of samples in a bicluster respectively. Let M_{opt} represent the set of implanted biclusters and M be the set of output biclusters of an algorithm. The score $S(M, M_{opt})$ represents the degree of similarity between extracted biclusters and the implanted biclusters, where as the score $S(M_{opt}, M)$ represents how well each of the true biclusters extracted by the bicluster algorithm.

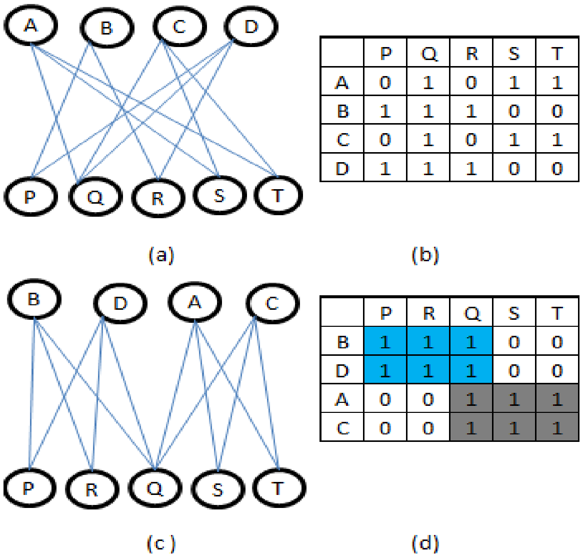
As per our experimental results, in case of high noise level for extraction of constant biclusters; BiCross, ISA and RMSBE shows high accuracies; BiMax and SAMBA perform moderately, and CC perform poorly. For coherent biclusters, BiCross has a comparable accuracy with RMSBE. In case of overlapped biclusters, BiCross is marginally affected by the overlap degree of the implanted biclusters.

5.2 Real Dataset

We have adopted the approach used by Zimmermann et al. [6] to evaluate the performance of BiCross with other algorithms for real gene expression data, provided by Gasch et al. [21]. In order to evaluate extracted biclusters based on Gene Ontology (GO) annotations [26] for their enrichment level, we have also used a web tool called FuncAssociate [27]. The adjusted significance scores (α) were computed using FuncAssociate and is shown in Figure 2. Based on this score, the results for BiCross is compared with other algorithms like BiMax, RMSBE, OSPM, SAMBA and CC.

5.3 Performance of BiCross

In this section we analyse the performance of our proposed BiCross algorithm. We have synthetically generated datasets with sizes ranging from 2000 × 100 to 100000 × 500 and implant constant biclusters in this matrix. Figure-3 illustrates the performance of both BiCross and RMSBE with respect to execution time for different size of dataset. As per our complexity analysis, the execution time of BiCross increase linearly with size of the dataset, while execution time for RMSBE increases at a much higher rate. This confirms the practical applicability of our proposed algorithm.



(a)

(b)

	P	Q	R	S	T
A	0	1	0	1	1
B	1	1	1	0	0
C	0	1	0	1	1
D	1	1	1	0	0

(c)

(d)

	P	R	Q	S	T
B	1	1	1	0	0
D	1	1	1	0	0
A	0	0	1	1	1
C	0	0	1	1	1

Figure 1: Applying the barycenter heuristic to a simple bipartite graph.

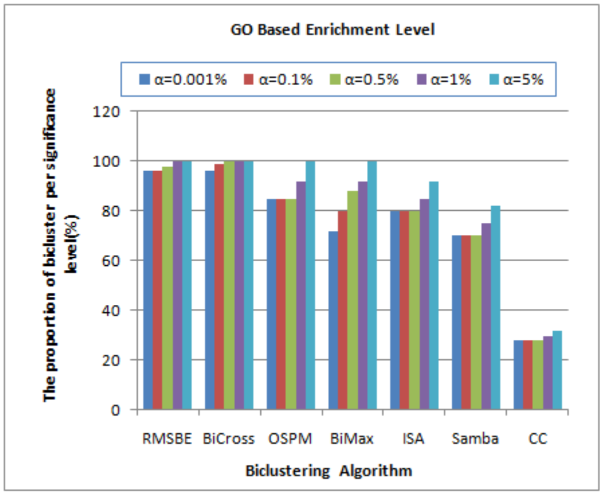


Figure 2: The proportion of GO based enriched biclusters

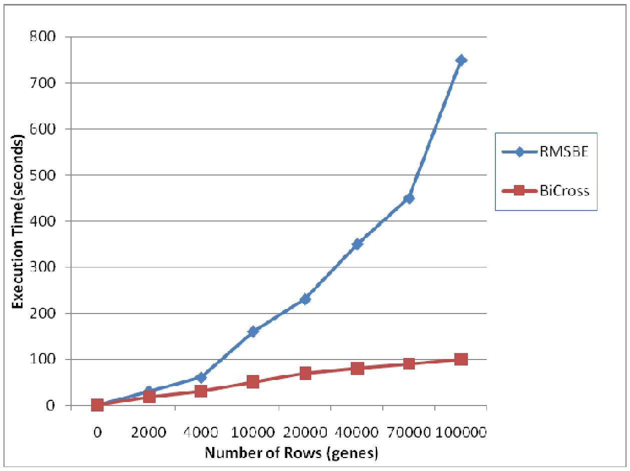


Figure 3: Performance of BiCross and RMSBE

```
$Algorithm - I$ :Weighted Crossing Minimization of a Bipartite Graph
Input : A Conditioned gene expression bipartite graph $G$
Output : Reordered bipartite graph $G'$ after application of WOLFCM
// Placing similar vertices of layer $V_1$ in partition $P_r$, where $0 \le r \le m$
// Vertices in $V_0$ are labeled with rank from 1 through $m$
// $A_u$ denote the set of all adjacent vertices of $u$ and $A_u[i]$ is the $i^{th}$
//    vertex label
1 : forall $u \in V_1$ do
2 : $R_{wsum} \leftarrow \sum_{i=1}^{|A_u|} w(uA_u[i])$ ;   $L_{wsum} \leftarrow 0$
3 : for $q : 1 \to |A_u| - 1$ do
4 : $r = A_u[q] - 1$
5 : $R_{wsum} \leftarrow R_{wsum} - w(uA_u[q])$
6 : if $L_{wsum} \ge R_{wsum}$ then goto step 13
7 : $L_{wsum} \leftarrow L_{wsum} + w(uA_u[q])$
8 : if adjacent nodes are consecutive then
9 :   if $L_{wsum} \ge (R_{wsum} - w(uA_u[q+1]))$
10 :        $r \leftarrow r + 1$
11 :        goto step 13
12 : endfor
13 : if $P_r$ is empty then $P_r \leftarrow P_r \cup \{u\}$
14 : else
15 :   if $v \in P_r$ then
16 :   if $\sum_{j=1}^{r} w(vA_v[j]) \times \sum_{j=r+1}^{m} w(uA_u[j]) \le \sum_{j=1}^{r} w(uA_u[j]) \times \sum_{j=r+1}^{m} w(vA_v[j])$ then
17 :     place $u$ to the right of $v$ in $P_r$
18 :   else
19 :     place $u$ to the left of $v$ in $P_r$
20 : endfor
21 : Reorder nodes in $V_1$ in the order $P_0, P_1, \ldots\ldots P_{m-1}$
```

6. CONCLUSION

We have proposed and implemented a biclustering technique, called BiCross, to extract constant, coherent and overlapped biclusters in the presence of noise. The technique employ weighted one layer free crossing minimization on a bipartite graph to identify a subset of genes and subset of conditions that form different types of biclusters. Our technique was found to work well for synthetic and real gene expression dataset, which is revealed by accuracy and performance measurement. The results also reveal that the technique outperform other conventional techniques in view of execution time.

```
Algorithm – II : Bicluster Extraction
Input : An $n \times m$ reordered matrix $D'$, minimum number
// of rows ($r$min), minimum number of columns ($c$ min),
// entropy threshold $\delta$, Bscore threshold $\beta$
Output : A set of biclusters
1 : srow $\leftarrow 0$ ; scol $\leftarrow 0$ ; overlap $\leftarrow$ false
2 : forall $i$ : srow $\to n - 2$ do
3 : forall $j$ : scol $\to m - 1$ do
4 : compute relative entropy between row $i$ and $i + 1$
      for column set $0 \to j$
5 : compute Bscore between adjacent columns
6 : if $i$ = srow then
7 :   if relative entropy $< \delta$ OR Bscore $< \beta$ then
8 :   reference column set (ref) $\leftarrow j$
9 : if $j = m - 1$ then row cluster (rcluster) $\leftarrow i$ and $i + 1$
10 : if $i >$ srow then
11 :   if relative entropy $< \delta$ OR Bscore $< \beta$ then
12 :   current column cluster (cccluster) $\leftarrow j$
13 : if $j = m - 1$ then
14 :   if ccluster contain almost same matching columns then
15 :     ref $\leftarrow$ ccluster , rcluster $\leftarrow i + 1$
16 :   if ccluster is empty then goto step 20
17 :   if ccluster contain more matching columns then
18 :     overlap $\leftarrow$ true , srow $\leftarrow i$ , scol $\leftarrow j$
19 :   if $i = n - 2$ goto step 20
20 : if |rcluster| $\ge$ rmin AND |ref| $\ge$ cmin then
21 : B $\leftarrow$ rcluster , ref ; clear rcluster , ccluster and ref
22 : if overlap = true then
23 :   $i \leftarrow$ srow ; $j \leftarrow$ scol ; overlap $\leftarrow$ false
24 : endfor
25 : endfor
26 : Return B
```

7. ACKNOWLEDGMENTS

The authors are grateful to the reviewers for their invaluable comments that have helped to improve the quality of this paper.

8. REFERENCES

[1] Cheng, Y. and Church, G. M. 2000. Biclustering of expression data. Proceedings of 8th International Conference on Intelligent Systems for Molecular Biology. 93-103.

[2] Tanay, A., Sharan, R. and Shamir, R. 2002. Discovering statistically significant biclusters in gene expression data. Bioinformatics, 18,S136-S144.

[3] Ben-Dor, A., Chor, B., Karp, R. and Yakhini,, Z. 2002. Discovering Local Structure in Gene Expression Data: The Order-Preserving Sub-matrix Problem. Proceedings of Sixth International Conference on Computational Molecular Biology (RECOMB '02). 49-57.

[4] Hussain, A. and Abdullah, A. 2006. A new biclustering technique based on crossing minimization. Neurocomputing Journal. 69,1982-1896.

[5] Wang, L. and Liu, X. 2007. Computing the maximum similarity bi-clusters of gene expression data. Bioinformatics. 23(1),50-56.

[6] Zimmermann, P., Wille, A., Buhlmann, P., Gruissem, W., Hennig, L., Thiele, L., Zitzler, E., Prelic A. and Bleuler, S. 2007. A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics. 23(1),50-56.

[7] Dhillon, I. S., Banerjee, A., Merugu, S. and Ghosh, J. 2005. Clustering with bregman divergences. Journal of Machine Learning Research. 6,1705-1749.

[8] Madeira, S. C. and Oliveira, A. L. 2004. Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Transactions on Computational Biology and Bioinformatics.1(1), 24-45.

[9] Garey, M. R. and Johnson, D. S. 1983. Crossing number is np-complete. SIAM Journal on Algebraic and Discrete Methods. 4,312-316.

[10] Tagawa, S., Sugiyama, K. and Toda, M. 1981. Methods for visual understanding of hierarchical system structures. IEEE Transaction on Systems, Man, and Cybernetics. 11(2),109-125.

[11] Eades, P. and Wormald, N. 1986.The Median heuristic for drawing 2-layers networks. Technical Report 69, Department of Computer Science, University of Queensland, Brisbane, Australia.

[12] Pach, J., Shahrokhi, F. and Szegedy, M. 1996. Applications of the crossing number. Algoritmica. 16, 111–117.

[13] Farhad Shahrokhi, Ondrej Sýkora, László A. Székely and Imrich V. 2001. On bipartite drawings and the linear arrangement problem. SIAM Journal on Computing. 30(6), 1773-1789.

[14] Jünger, M. and Mutzel, P. 1997. 2-layer straightline crossing minimization, Performance of exact and heuristic algorithms, Journal of Graph Algorithms and Applications. 1(1),1-25.

[15] Brown, P. O., Alter, O. and Botstein,D. 2000. Singular value decomposition for genome-wide expression data processing and modeling, PNAS. 97,10101-10106.

[16] Troyannskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D. and Altman, R. 2001. Missing value estimation methods for dna microarrays. Bioinformatics. 17(6), 1-6.

[17] Tchagang, A. B. and Tewfik, A.H. 2005. Robust biclustering algorithm (roba) for dna microarray data nalysis, Proceedings of IEEE Workshop on Statistical Signal Processing.

[18] Yang, Y. and Webb, G.I. 2002. A comparative study of discretization methods for naïve-baysian classifiers. Proceedings of Pacific Rim Knowledge Acquisition Workshop, National Center of Sciences, Tokyo, Japan.

[19] Devore, J. L. 1995. Probability and Statistics for Engineering and the Sciences, Duxbury Press, 4th edition.

[20] Munoz, X., Unger, W. and Vrto, I. 2002. One sided crossing minimization is np-hard for sparse graphs. International Symposium on Graph Drawing, London, UK. 115-123, Springer-Verlag.

[21] Gasch, A.P. 2000. Genomic expression programs in the response of yeast cells to environmental changes. Molecular Biology Cell. 11,4241-4257.

[22] Kriegel, H., Kr□ger, P. and Zimek, A. 2009. Clustering High-Dimensional Data: A Survey on Subspace Clustering, Pattern-Based Clustering, and Correlation Clustering. ACM transactions on knowledge discovery from data. 3(1), Article 1.

[23] Bergmann, S., Ihmels, J. and Barkai, N. 2004. Defining transcription modules using large-scale gene expression data. Bioinformatics, 20(13),1993-2003.

[24] Maron-Katz, A., Sharan, R. and Shamir, R. 2003. Click and expander: A system for clustering and visualizing gene expression data. Bioinformatics. 19(14),1787-1799.

[25] Prelic, A., Zimmermann, P., Barkow, S., Bleuler, S. and Zitzler, E. 2006. Bicat: a biclustering analysis toolbox. Bioinformatics. 22(10),1282-1283.

[26] Gene Ontology Consortium (2000). Gene ontology: tool for the unification of biology. *Nat. Genet.*. 25,25-29.

[27] Berriz, G., Bryant, O., Sander, C. and Roth, F. 2003. Charactering gene sets with FuncAssociate, Bioinformatics, 22,1282-1283.

[28] Cover, T. M. and Thomas, J. A. 2006. Elements of Information Theory. Wiley.

[29] Han, J. and Kamber, M. 2011. Data Mining: Concepts and Techniques. Elsevier.