American Journal of Bioinformatics Research 2012, 2(4): 40-46
DOI: 10.5923/j.bioinformatics.20120204.02

BiRange:An Efficient Framework for Biclustering of Gene Expression Data Using Range Bipartite Graph

Suvendu Kanungo[1,*], **Gadadhar Sahoo**[2], **Manoj Madhava Gore**[3]

[1]Department of Computer Sc., BIT Mesra (Allahabad Campus), Allahabad, 211010, India
[2]Department of IT, BIT Mesra, Ranchi, 835215, India
[3]Department of Computer Sc. & Engg., MNNIT, Allahabad, 211004, India

Abstract Biclustering is a vital data mining tool which is commonly employed on microarray data sets for analysis task in bioinformatics research and medical applications. There has been extensive research on biclustering of gene expression data arising from microarray experiment. This technique is an important analysis tool in gene expression measurement, when some genes have multiple functions and experimental conditions are diverse. In this paper, we introduce a new framework for biclustering of gene expression data. The basis of this framework is the construction of a range bipartite graph for the representation of 2-dimensional gene expression data. We have constructed this range bipartite graph by partitioning the set of experimental conditions into two disjoint sets. The key benefit of this representation is that, it leads to a compact representation of all similar value ranges between experimental conditions. Based on this problem formulation, an efficient algorithm is proposed that searches for constrained maximal cliques in this range bipartite graph, in order to extract a set of biclusters. Our technique is scalable to practical gene expression data and can produce different types of biclusters amid noise. The experimental evaluation of this technique also reveals its accuracy and effectiveness with respect to noise handling and execution time in comparison to other similar techniques.

Keywords Microarray, Biclustering, Gene expression data, Bipartite Graph

1. Introduction

Clustering is commonly used to reveal biological meaningful pattern in data arising from microarray experiment, which is called gene expression data. Further, it is the most common tool for interaction identification, as similar objects form a cluster. Clustering is unsupervised classification[1], also known as cluster analysis, which discovers grouping(s) of a set of patterns, objects or points. It is prevalent in any discipline that involves interaction identification. Unfortunately, clustering is difficult for most data sets due to its diverse shapes, densities, sizes and background noise. Gene expression data are usually arranged in a 2-dimensional matrix, where rows represent genes and columns represent samples or experimental conditions. Each element of this matrix represents the expression level of a gene under a specific condition, and is represented by a real number. In this case, genes that are similar may share a common biological pathway and the groupings of predictivegenes can be of interest to biologist. Conventional clus ter ing techniques are based on similarity between genes across all samples or experimental conditions. However, genes may be co regulated under some specific experimental conditions and shows weak similarity beyond these conditions. Therefore, a group of genes forms cluster under a subset of conditions. This technique of two-way clustering referred to as biclustering, in which both genes and conditions are clustered simultaneously.

2. Related Work

Several biclustering algorithms have been proposed in different application scenarios but we concentrate on graph theoretic approaches. In order to extract biclusters, these algorithms usually employ heuristic or probabilistic model. An illustrative discussion on many of these algorithms can be found in[4,5].

Cheng and Church[2] 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 produce 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.

[*] Corresponding author:
skanungo.bit@rediffmail.com (Suvendu Kanungo)

Published online at http://journal.sapub.org/ bioinformatics

Tanay et al.[6] 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 performs 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.

Ahsan and Amir[3] 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 produce coherent biclusters, as this method use a static discretization of the input data matrix.

Waseem and Asfaq[7] proposed cHawk, to identify biclusters with the help of crossing minimization paradigm. This method employs the barycenter heuristic to arrange vertices in both layers of a bipartite graph. The similarity test is done based on bregman divergence. This approach is similar to our approach as we also employ bipartite graph for representation of gene expression data. The time complexity of this technique is $O(dnm)$, where n and m are the number of rows and columns of the input data matrix and d is the average degree of overlap among biclusters, which is slower than our approach.

Wang and Liu[8] 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 $O(nm(m+n)^2)$, 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.[9] 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.

Bergmann et al.[10] 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.

Zhao and Zaki[11] proposed Tricluster, for mining coherent clusters in 3-dimensional gene expression data sets. They construct a range multigraph and then searches for constrained maximal cliques in this multigraph, in order to extract a set of biclusters. However, they do not address the issue of noisy data, where as our approach can effectively handle noisy data.

3. Model Formulation

Let $J = \{g_0, g_1, \cdots, g_{n-1}\}$ be a set of n genes and $C = \{c_0, c_1, \cdots, c_{m-1}\}$ be a set of m experimental conditions. Microarray data-set is a real valued $n \times m$ expression matrix $D = J \times C = \{d_{ij}\}$ 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 experimental condition C_j. A bicluster corresponds to a sub matrix that exhibits some coherent tendency. Let B be a sub matrix of dataset D i.e Bicluster $B = X \times Y = \{b_{ij}\}$ where $X \subseteq J$ and $Y \subseteq C$, 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 S be the set of all biclusters that satisfy the given homogeneity conditions, then $B \in S$ is called a maximal bicluster iff there doesn't exist another bicluster $B' \in S$ such that $B \subset B'$.

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

B will be a valid bicluster iff it is a maximal bicluster satisfying the following conditions:

Let us consider $\mu_{g_i} = \sqrt{\prod_{j=1}^{2} b_{ij}}$ be the geometric mean between two specified column values for a given row and $w_i = \frac{\mu_{g_i}}{\sum_i \mu_{g_i}}$ be the weight of the row for this specified two column values.

Let us consider $r_i = w_i \times |b_{iq} - b_{ip}|$ and $r_j = w_j \times |b_{jq} - b_{jp}|$ be the weighted difference of two column values for a given row i and j respectively. We need that $max\,(r_i, r_j) - \min(r_i, r_j) \leq \rho$; where ρ is the multiple of maximum weight in the corresponding gene-set i.e $\varepsilon \times \max(w_i)$.

We also need that $|X| \geq \sigma_x$ and $|Y| \geq \sigma_y$, where σ_x and σ_y denote minimum cardinality thresholds for each dimension.

We consider an edge as valid, when the weighted difference range for a condition pair satisfies the threshold value so that it generates a gene-set. In order to produce large enough clusters, the minimum size constraints i.e σ_x, and σ_y are imposed. B is a scaling bicluster if $b_{iq} = \alpha_i b_{ip}$ and $b_{jq} = \alpha_j b_{jp}$; and $\alpha_{i-}\alpha_j \leq \delta$, 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 $\beta_i - \beta_j \leq \delta$, 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.

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. $V_0 \cap V_1 = \emptyset$).

We consider weighted bipartite graph $G(V_0, V_1, E, W)$ with $W = (w_{ij})$, where $w_{ij} \geq 0$ denotes the weight of the edge (i, j) between vertices i and j.

3.1. Preprocessing

Gene expression data is usually noisy and may contain missing values. Illustrative discussion on prediction of missing value can be found in[12, 13]. Therefore it is essential to condition the data before applying the clustering algorithm. In order to handle missing values, we have adopted the approach used in[14] i.e. replacing all missing values by zero. Before normalizing the dataset, data beyond a threshold value (three standard deviation), has been temporarily removed to reduce the effect of outliers in the data. Then, the gene expression data is transformed using z-score standardization, where the transformed variables have a mean of 0 and variance of 1. Finally, the temporarily removed outliers that are below the mean value are replaced by the minimum value, where as the outliers above the mean values are replaced by the maximum value of the final normalized data. In order to handle outlier efficiently, we have partitioned the normalized condition 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[15]. In this approach, we partition the condition column 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 condition column has been partitioned into required number of intervals. The number of intervals r for each condition depends on the data size n. Further, to have balanced partition, it is assumed that $r = 2^k$, where k is a positive integer and $r^2 \times 35 \leq n$, where 35 is the minimum sample size for large sample procedures[16]. 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.

3.2. Constructing Weighted Range Bipartite Graph

For a given dataset D, the minimum size threshold, σ_x and σ_y, and the maximum weighted difference threshold ρ, let c_u and c_v be any two condition columns of D and let $r_x^{uv} = w_x \times |d_{xu} - d_{xv}|$ be the weighted difference of the expression values of gene g_x in columns c_u and c_v such that $u < v$, where $x \in [0, n-1]$. In order to incorporate the idea of mutual importance between two columns, we have computed the weight of all rows for specified column pairs. A difference range is defined as an interval of difference values $[r_l, r_h]$, with $r_l < r_h$. Let $J([r_l, r_h]) = \{g_x : r_x^{uv} \in [r_l, r_h]\}$ be the set of genes, whose difference w.r.t. columns c_u and c_v lie in the given weighted difference range. A difference range is called valid iff $\max(r_h, r_l) - \min(r_h, r_l) \leq \rho$, where ρ is the multiple of maximum row weight in the corresponding gene set. Normally, for microarray experiment data, genes and conditions are represented by V_1 and V_2 vertex sets respectively, and the edge weight w_{ij} represents the response of i^{th} gene to j^{th} condition. However, in order to have a very compact representation, in this paper, we construct the weighted undirected bipartite graph by partitioning the condition set into two disjoint sets called upper layer (V_1) and lower layer (V_2). The conditions that do not have any data values are not considered in the formation of disjoint sets. Here, each edge in the range bipartite graph has associated with it the rank and gene-set corresponding to the weighted difference range on that edge. Different bipartite graphs emerged for different threshold value, which is the multiple of maximum weight value in the corresponding gene-set. Consequently, we will have different types of biclusters. We have given priority to all valid edges that have large number of genes in the gene-set, by assigning rank (R_e) to these edges. Ranks have been assigned on the basis of total number of genes in the gene-set i.e. $|J([r_l, r_h])|$. The gene-sets having highest cardinality have been assigned rank 1, the second highest rank 2, the third highest rank 3 and so on. The inclusion and deletion of edges depends upon the value of R_e and order in which we process these ranks and conditions. Let E' and J' be the set all valid edges and gene-sets respectively, in ascending order of rank values.

Table 1. Example of Microarray Dataset

	c0	c1	c2	c3	c4
g0	1	2	3	0	0
g1	0	0	0	4	5
g2	1	2	3	0	0
g3	0	0	0	4	5
g4	1	2	3	0	0

Figure 1. illustrates the steps for computing the weighted difference range for columns c_0 and c_1 in Table 1. Let $\rho = 0.33$, which is the maximum row weight in the gene-set, and considering $\sigma_x = 3$, $\sigma_y = 2$, then there is only one valid weighted difference range[0.33, 0.33] and the corresponding gene-set in sorted order is $J_{c_0, c_1}([0.33, 0.33]) = \{g_0, g_2, g_4\}$. Similarly, we compute weighted difference range for other experimental conditions. In this case, the number of valid ranges depends on the value of ρ. For the sorted difference values, we find all valid weighted differ-

ence ranges for all pair of columns $c_u, c_v \in C$. Further, for simplicity, we have not considered rows with weight 0.

$\mu_g(c_0, c_1)$	1.414	1.414	1.414
Row	g_0	g_2	g_4

(a) Geometric Mean between c_0 and c_1

$w_i(c_0, c_1)$	0.33	0.33	0.33
Row	g_0	g_2	g_4

(b) Weight of Row i between c_0 and c_1

$c_0 - c_1$	1	1	1
Row	g_0	g_2	g_4

(c) Difference between c_0 and c_1

$w_i \times \lvert c_0 - c_1 \rvert$	0.33	0.33	0.33
Row	g_0	g_2	g_4

(d) Weighted Difference between c_0 and c_1

Figure 1. Weighted difference range for c_0 and c_1

Here we may have overlapping of different ranges. The algorithm for partitioning condition set into V_1 and V_2, for construction of bipartite graph is given in Algorithm I. From Table 1, a maximal weighted range bipartite graph is constructed (Figure 2). Let C' be the set of columns with missing value in each row. Let C'' be the set of conditions such that $C'' = C - C'$. We have taken weight of the edge as rank for the corresponding gene-set. This algorithm gives priority to the edges having large rank in order to compensate the deletion of few valid edges, while partitioning the condition set into two disjoint sets. If there is a tie among rank values, then we randomly select an edge and start constructing the bipartite graph. As we deal with noisy data, additive and multiplicative methods of finding clusters may not always lead to good results. Therefore, instead of comparing two column values independently[11], we have computed weight of each row for any two specified column values. We build bipartite graph model of data, after properly conditioning the input data.

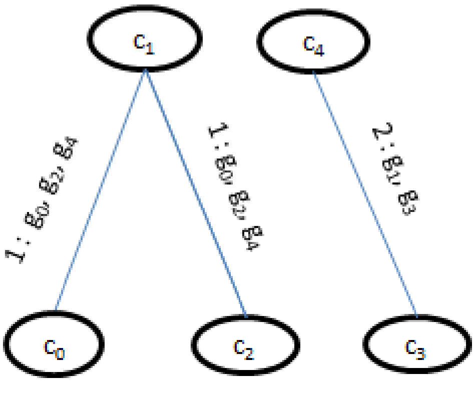

Figure 2. Weighted Range Bipartite Graph G′

Algorithm I: Partitioning of Condition Set C''
Input: E', C''
Output: Creation of two disjoint sets V_1 and V_2

Initialization: $V_1 = \emptyset$,
$V_2 = \emptyset$

```
while $V_1 \cup V_2 \neq C''$
    for each valid edge $(c_u, c_v) \in E'$ do
        if $c_u$ and $Adj(c_v) \notin V_1$ then
            insert $c_u$ and $Adj(c_v)$ in $V_1$
        endif
        if $c_v$ and $Adj(c_u) \notin V_2$ then
            insert $c_v$ and $Adj(c_u)$ in $V_2$
        endif
    endfor
endwhile
```

3.3. Experimental Setup

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. The accuracy and performance of this algorithm is evaluated using synthetically generated dataset and real dataset. For synthetic data generation, a technique parallel to a methodology proposed in[9] is adopted whereas for real dataset, the model organism Yeast *Saccharomyces Cerevisiae* dataset is considered, since the yeast GO annotations are more extensive compared to other organisms. This dataset is provided by Gasch et al.[17], which contains 2,993 genes and 173 different stress conditions.

4. Bicluster Extraction and Evaluation

4.1 Bicluster Extraction

The weighted range bipartite graph is constructed by partitioning the set of experimental conditions $\boldsymbol{C''}$ into two disjoint sets $\boldsymbol{V_1}$ and $\boldsymbol{V_2}$. Each edge of this graph is associated with a rank value and the corresponding gene-set. The algorithm for extraction of biclusters from this graph by employing depth first search technique is given in Algorithm II. This algorithm requires the value of $\boldsymbol{\sigma_x}$, $\boldsymbol{\sigma_y}$ and the weighted bipartite graph $\boldsymbol{G'}$ as its input parameter. The output of this algorithm is a set of biclusters $\boldsymbol{S}$ and the number of such biclusters depends upon the dataset.

For example, let us consider the value of input parameters $\boldsymbol{\sigma_x}$=3 and $\boldsymbol{\sigma_y}$=2. The algorithm starts searching the graph $\boldsymbol{G'}$ at a valid edge having highest rank value. In this case, it starts with the valid edge $(\boldsymbol{c_0}, \boldsymbol{c_1})$ and get the corresponding reference cluster $\{\boldsymbol{g_0}, \boldsymbol{g_2}, \boldsymbol{g_4}\} \times \{\boldsymbol{c_0}, \boldsymbol{c_1}\}$. Then, other adjacent valid edges are explored that leads to the formation of final bicluster $\{\boldsymbol{g_0}, \boldsymbol{g_2}, \boldsymbol{g_4}\} \times \{\boldsymbol{c_0}, \boldsymbol{c_1}, \boldsymbol{c_2}\}$. If a new edge does not have the sufficient number of genes and conditions, the reference bicluster is declared as the final bicluster, since this is maximal and satisfies all required conditions. Further, other valid edges are searched that leads to the formation of final bicluster $\{\boldsymbol{g_1}, \boldsymbol{g_3}\} \times \{\boldsymbol{c_3}, \boldsymbol{c_4}\}$. Here both the biclusters satisfies the minimum threshold value.

Algorithm II: Bicluster Extraction
Input: $G', \sigma_x, \sigma_y, C'', J, J', E'$
Output: S

```
Initialization: $S = \emptyset$,
Call $BEXTRACT\ (B = J \times \emptyset)$
$BEXTRACT(B = P \times Q)$
  if $|B \cdot Q| \geq \sigma_y$ then
    if $B' \notin S$ such that $B \subset B'$ then
      if $B'' \subset B$ then
        remove $B''$
        $S \leftarrow S + B$
      endif
    endif
  endif
  foreach $c_u \in C''$ do
    $B^R \leftarrow B$
    $B^R \leftarrow B^R \cdot Q + c_u$
    forall edges $(c_u, c_v) \in E'$ do
      if $|J'(c_u, c_v) \cap B \cdot P| \geq \sigma_x$ then
        $B^R \cdot P = J'(c_u, c_v) \cap B \cdot P$
      endif
      if $|B^R \cdot P| \geq \sigma_x$ then
        $BEXTRACT\ (B^R)$
      endif
    endfor
  endfor
  Return $S$
```

4.2. Evaluation

For evaluation purpose, we need to compare our algorithm with other biclustering algorithms. As different biclustering algorithms deals with different problem formulations and clustering criteria, it is difficult to have a comparative study among such algorithms. Therefore, these algorithms work efficiently in certain situation and perform poorly in others. In view of this problem, we need to provide a common setting for such algorithms so that we can perform a fair comparative study. Our main focus lies on validating our proposed algorithm for extraction of constant, coherent and overlapped biclusters from noisy gene expression data with high accuracy in comparison to other similar algorithms. In view comparison, similar algorithms like CC[2], ISA[10], cHawk[7], SAMBA[6], RMSBE[8] and BiMax[9] are considered. We have used the Bicluster Analysis Toolbox (BicAT) developed by Prelic et al.[18] for implementation of BiMax, CC and ISA. Further, for implementation of SAMBA, EXPANDER developed by Maron-Katz et al.[19] is used and RMSBE implementation was downloaded from[20].

4.2.1. Complexity Analysis

Since we require to evaluate all pair of conditions, compute the weight of each gene, and find the weighted difference range to get the corresponding gene-set, the range bipartite graph construction step would take time $O(nm^2)$. Here, we have considered the experimental conditions as vertices for the bipartite graph. These vertices are partitioned into two disjoint sets on the basis of rank of valid edges. This way of constructing bipartite graph leads to deletion of some valid edges and as a result, fewer edges need to be processed in comparison to other graph approach for solving the same problem. Therefore, the running time can be significantly reduced for this representation of gene expression data using a bipartite graph. The bicluster extraction step depends on the value of input parameters and datasets. This step is most expensive as there can be exponential number of clusters. Since the experimental conditions are only considered as vertices, and some uninteresting edges may be pruned, the depth of the search in weighted range bipartite graph to extract biclusters is likely to be small in comparison to multigraph representation[11].

4.2.2. Synthetic Dataset

In order to evaluate implanted constant, coherent and overlap biclusters in synthetic data, we have used the technique proposed by Zimmermann et al.[9]. 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 val-ues 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 × 100to 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. The parameter setting for different algorithms is shown in Table 2. In order to validate the accuracies of different algorithms, we apply the gene match score proposed by Zimmermann et al.[9]. 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:

$$S_J(M_1, M_2) = \frac{1}{|M_1|} \sum_{(J_1, S_1) \in M_1} \max_{(J_2, S_2) \in M_2} \frac{|J_1 \cap J_2|}{|J_1 \cup J_2|} \qquad (1)$$

where J and S are set of genes and a set of conditions 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.

Figure 3 illustrates the experimental results with respect to the accuracy evaluation of constant biclusters. As per the experimental results, in case of high noise level for extraction of constant biclusters; BiRange along with cHawk, ISA and RMSBE shows high accuracies; BiMax and SAMBA perform moderately, and CC perform poorly. Figure 4 illustrates the experimental results with respect to the accuracy evaluation of coherent biclusters. For coherent biclusters, BiRange has a comparable accuracy with RMSBE and cHawk. Figure 5 illustrates the experimental results with respect to the accuracy evaluation of overlapped biclusters. In case of overlapped biclusters, BiRange is marginally affected by the overlap degree of the implanted biclusters.

Table 2. Parameter Settings for Different Biclustering Techniques

S. N.	Algorithm	Setting
1	BiRange	$\rho \leq 0.5, I = 5$
2	SAMBA	$D = 40, N_1 = 6, N_2 = 6,$ $k = 20, L = 30$
3	CC	$\delta \leq 0.5, \alpha = 1.2$
4	cHawk	$\delta = 0.5, I = 5$
5	BiMax	$Min.\ number\ of\ chips\ or\ genes = 12$
6	RMSBE	$\alpha = 0.4, \beta = 0.5, \gamma = \gamma_c = 1.2$
7	ISA	$t_g = 2.0, t_c = 2, seeds = 500$

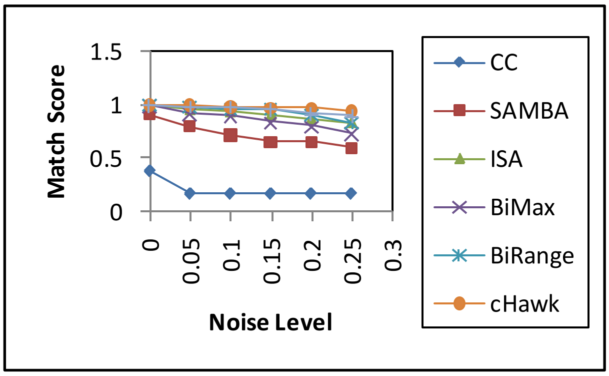

Figure 3. Accuracy of Constant Biclusters

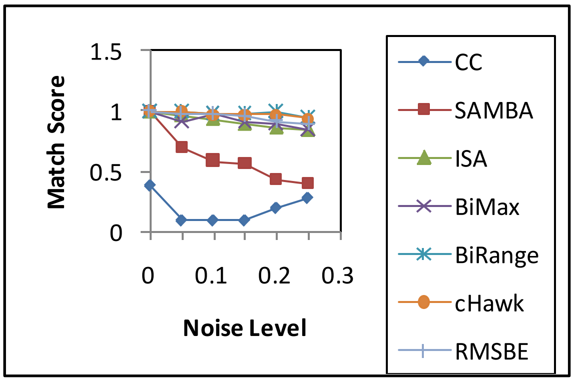

Figure 4. Accuracy of Coherent Biclusters

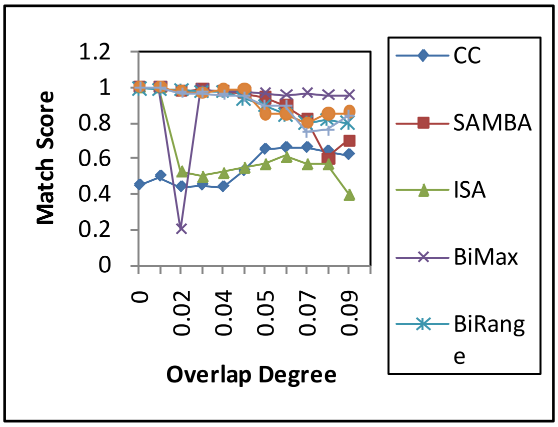

Figure 5. Accuracy of Overlapped Biclusters

4.2.3. Real Dataset

For real gene expression dataset, provided by Gasch et al.[17], the performance of the proposed technique BiRange is evaluated with respect to other similar algorithms based on the methodology used by Zimmermann et al.[9]. In order to evaluate extracted biclusters for their enrichment level based on Gene Ontology (GO) annotations[21], a web tool called *FuncAssociate*[22] is also used. The adjusted significance scores (α) were computed using *FuncAssociate* and is shown in Figure 6. The experimental results for BiRange is compared with other algorithms like BiMax, RMSBE, cHawk, SAMBA, ISA and CC based on this significance score.

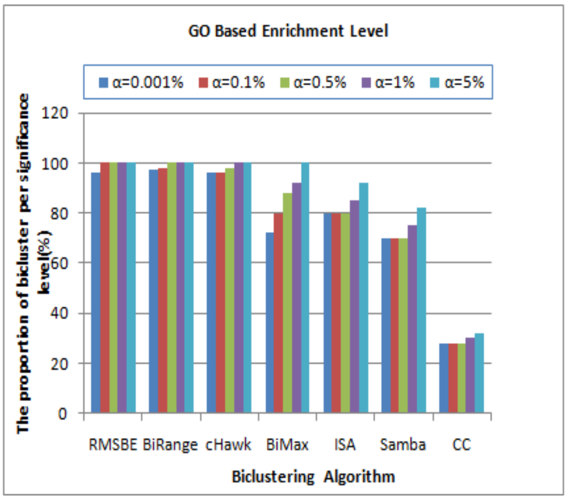

Figure 6. Proportion of GO Based Enriched Biclusters

4.2.4. Performance of BiRange

In this section the performance of the proposed BiRange algorithm is analyzed. We have synthetically generated datasets with sizes ranging from 2000 × 100 to 100000 × 500 and implant constant biclusters in this matrix. Figure 7 illustrates the performance of BiRange, cHawk and RMSBE with respect to execution time for different size of dataset. As per our complexity analysis, the execution time of BiRange increases approximately linearly with the number of genes in a cluster, while execution time for RMSBE increases at a much higher rate. This confirms the practical applicability of our proposed algorithm.

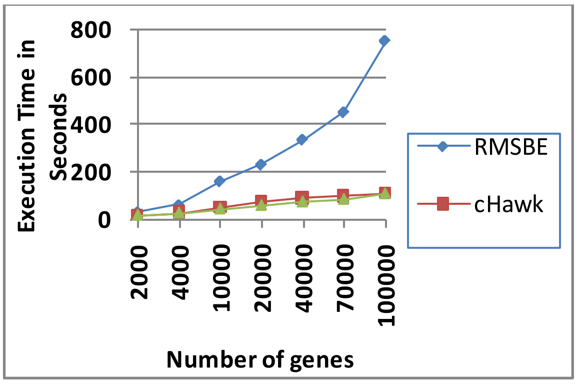

Figure 7. Performance of BiRange, cHawk and RMSBE

5. Conclusions

We have represented the gene expression data using a weighted range bipartite graph, by computing the weight of each gene for a specified condition pair. For the construction of bipartite graph, the set of experimental conditions are partitioned into two disjoint sets based on the rank of valid edges. The rank value is computed from the cardinality of the corresponding gene-set, which is associated with the valid edge. The motivation of considering conditions as vertices of the bipartite graph is due to large number of genes in real gene expression data. The proposed algorithm has been evaluated for synthetic as well as real microarray data. The experimental results reveal the effectiveness of our approach over other biclustering approaches with respect to time.

ACKNOWLEDGEMENTS

We would like thank the anonymous referees for their invaluable comments and suggestions that greatly improved the quality and presentation of this paper.

REFERENCES

[1] J. Han, and M. Kamber, Data Mining: Concepts and Techniques, Elsevier, 2001

[2] Y. Cheng and G.M. Church, "Biclustering of expression data,"in 8th Int'l conference of Intelligent Systems for Molecular Biology, pp. 93-103, 2000

[3] A. Hussain and A. Abdullah, "A new biclustering technique based on crossing minimization," Neurocomputing Journal, vol. 69, pp. 1982–1986, 2006

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

[5] R. Xu and D. Wunsch "Survey of clustering algorithms,"IEEE trans. on Neural Networks, vol. 16(3), 645-678, 2005

[6] A. Tanay, R. Saran and R. Samir, "Discovering statistically significant biclusters in gene expression data analysis, vol. 18, pp. S136-S144, 2002

[7] W. Ahmad and A. Khokhar, "cHawk: An efficient biclustering algorithm based on bipartite graph crossing minimization,"ACM, VLDB, 2007

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

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

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

[11] L. Zhao and M. J. Zaki, "TRICLUSTER:An effective algorithm for mining coherent clusters in 3D microarray data, SIGMOD, 2005

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

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

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

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

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

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

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

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

[20] L. Wang and X. Liu, "MSBE software download", http://www.cs.cityu.edu.hk/linuxw/msbe/help.html, 2007

[21] Gene Ontology Consortium, "Gene ontology: tool for the unification of biology", Natural Genetics, vol. 25, pp. 25-29. 2000

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