International Journal of Computer and Communication Technology

Volume 3 | Issue 3 Article 6

July 2012

Graph Theoretic Techniques for Clustering and Biclustering gene expression data.

Prangyaparamita Mohapatra
prangyaparamita.mohapatra@gmail.com

Tripti Swarnkar
Deptt. Of CA, SOA University Bhubaneswar, tripti_sarap@yahoo.com

Graph Theoretic Techniques for Clustering and Biclustering gene expression data.

Prangyaparamita Mohapatra, Tripti Swarnkar
prangyaparamita.mohapatra@gmail.com,tripti_sarap@yahoo.com

Abstract: **DNA microarray technology has made it possible to simultaneously monitor the expression levels of thousands of genes during biological processes and across collections of related samples. However, the large number of genes and the complexity of biological networks greatly increase the challenges of comprehending and interpreting the resulting mass of data, which often consists of millions of measurements. A first step toward addressing this challenge is the use of clustering techniques, which is essential in the data mining process to reveal natural structures and identify interesting patterns in the underlying data. Cluster analysis seeks to partition a given data set into groups based on specified features so that the data points within a group are more similar to each other than the points in different groups. Many conventional clustering algorithms have been adapted or directly applied to gene expression data, and also new algorithms have recently been proposed specifically aiming at gene expression data. These clustering algorithms have been proven useful for identifying biologically relevant groups of genes and samples. A large number of clustering approaches have been proposed for the analysis of gene expression data obtained from microarray experiments. However, the results of the application of standard clustering methods to genes are limited. These limited results are imposed by the existence of a number of experimental conditions where the activity of genes is uncorrelated. A similar limitation exists when clustering of conditions is performed. For this reason, a number of algorithms that perform simultaneous clustering on the row and column dimensions of the gene expression matrix have been proposed to date. This simultaneous clustering, usually designated by biclustering, seeks to find sub-matrices that are subgroups of genes and subgroups of columns, where the genes exhibit highly correlated activities for every condition. This type of algorithms has also been proposed and used in other fields, such as information retrieval and data mining. In this paper, we first briefly introduce the concepts of microarray technology and discuss the basic elements of clustering on gene expression data. Then, we present specific challenges pertinent to each clustering category and introduce several representative approaches.**

Keywords: **Gene expression; Clustering; Bi-clustering; Microarray analysis**

1 INTRODUCTION

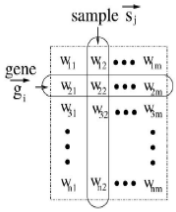
1.1 Introduction to Microarray Technology

Compared with the traditional approach to genomic research, which has focused on the local examination and collection of data on single genes, microarray technologies have now made it possible to monitor the expression levels for tens of thousands of genes in parallel. The two major types of microarray experiments are the cDNA microarray and oligonucleotide arrays (abbreviated oligo chip). Despite differences in the details of their experiment protocols, both types of experiments involve three common basic procedures:

- *Chip manufacture.*
- *Target preparation, labeling, and hybridization*
- *The scanning process*

1.1.1 Preprocessing of Gene Expression Data

A microarray experiment typically assesses a large number of DNA sequences.In this paper, we will focus on the cluster analysis of gene expression data without making a distinction among DNA sequences, which will uniformly be called “genes”. Similarly, we will uniformly refer to all kinds of experimental conditions as “samples” if no confusion will be caused. A gene expression data set from a microarray experiment can be represented by a real-valued expression matrix $M=\{w_{ij} \mid 1 \le i \le n, 1 \le j \le m\}$(Fig. 1a), where the rows $(G=\{g_1,\ldots,g_n\})$form the expression patterns of genes, the columns $(S=\{s_1,\ldots,s_n\})$represent the expression profiles of samples, and each cell w_{ij} is the measured expression level of gene i in sample j. Fig. 1b includes some notation that will be used in the following sections.



(a)

n	number of genes
m	number of samples
M	a gene expression matrix
w_{ij}	each cell in a gene expression matrix
$\vec{g_i}$	a gene
$\vec{s_j}$	a sample
$G, G', G_0, ...$	a set of genes
$S, S', S_0, ...$	a set of samples

(b)

Fig. 1.1 (a) A gene expression matrix. (b) Notation in this paper.

1.1.2 Applications of Clustering Gene Expression Data

Clustering techniques have proven to be helpful to understand gene function, gene regulation, cellular processes, and subtypes of cells. Genes with similar expression patterns (coexpressed genes) can be clustered together with similar cellular functions. Furthermore, coexpressed genes in the same cluster are likely to be involved in the same cellular processes, and a strong correlation of expression patterns between those genes indicates coregulation. Searching for common DNA sequences at the promoter regions of genes within the same cluster allows regulatory motifs specific to each gene cluster to be identified and cis-regulatory elements to be proposed. The inference of regulation through the clustering of gene expression data also gives rise to hypotheses regarding the mechanism of the transcriptional regulatory network. Finally, clustering different samples on the basis of corresponding expression profiles may reveal subcell types which are hard to identify by traditional morphology-based approaches

1.1.3 Introduction to Clustering Techniques

In this section, we will first introduce the concepts of clusters and clustering. We will then divide the clustering tasks for gene expression data into three categories according to different clustering purposes. Finally, we will discuss the issue of proximity measure in detail.

.

1.1.4 Clusters and Clustering

Clustering is the process of grouping data objects into a set of disjoint classes, called *clusters*, so that objects within a class have high similarity to each other, while objects in separate classes are more dissimilar. Clustering is an example of *unsupervised classification*. "Classification" refers to a procedure that assigns data objects to a set of classes. "Unsupervised" means that clustering does not rely on predefined classes and training examples while classifying the data objects. Thus, clustering is distinguished from pattern recognition or the areas of statistics known as discriminant analysis and decision analysis, which seek to find rules for classifying objects from a given set of preclassified objects.

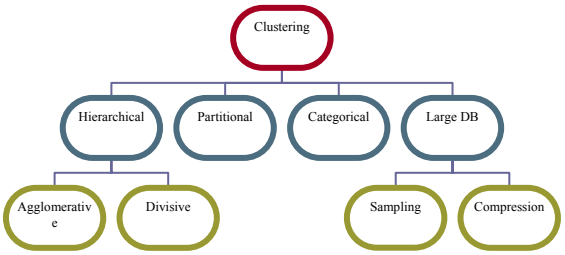


Fig. 1.2 Classification of clustering algorithms.

1.2.1 Categories of Gene Expression Data Clustering

One of the characteristics of gene expression data is that it is meaningful to cluster both genes and samples. On one hand, co expressed genes can be grouped in clusters based on their expression patterns. In such ***gene-based*** clustering, the genes are treated as the objects, while the samples are the features. On the other hand, the samples can be partitioned into homogeneous groups. Such ***sample-based*** clustering regards the samples as the objects and the genes as the features. Some clustering algorithms, such as K-means and hierarchical approaches, can be used both to group genes and to partition samples.Current thinking in molecular biology holds that only a small subset of genes participates in any cellular process of interest and that a cellular process takes place only in a subset of the samples. This belief calls for the ***subspace clustering*** to capture clusters formed by a subset of genes across a subset of samples. Thus, we may have to adopt very different computational strategies in the three situations.

2 CLUSTERING ALGORITHMS

As we mentioned in Section 1.2.2, gene expression matrix can be analyzed in two ways. For gene-based clustering, genes are treated as data objects, while samples are considered as features. Conversely, for sample-based clustering, samples serve as data objects to be clustered, while genes play the role of features. The third category of cluster analysis applied to gene expression data, which is subspace clustering, treats genes and samples symmetrically such that either genes or samples can be regarded as objects or features. Gene-based, sample-based, and subspace clustering face very different challenges, and different computational strategies are adopted for each situation.

2.1 Gene-Based Clustering

In this section, we will discuss the problem of clustering genes based on their expression patterns.

2.1.1 Challenges of Gene Clustering

Due to the special characteristics of gene expression data, and the particular requirements from the biological domain, gene-based clustering presents several new challenges and is still an open problem.

- First, cluster analysis is typically the first step in data mining and knowledge discovery.
- Second, due to the complex procedures of microarray experiments, gene expression data often contains a huge amount of noise.
- Third, our empirical study has demonstrated that gene expression data are often "highly connected", and clusters may be highly intersected with each other or even embedded one in another .
- Finally, users of microarray data may not only be interested in the clusters of genes, but also be interested in the relationship between the clusters, and the relationship between the genes within the same cluster .

2.1.2 Hierarchical Clustering

Hierarchical clustering generates a hierarchical series of nested clusters which can be graphically represented by a tree, called *dendrogram*. The branches of a dendrogram not only record the formation of the clusters but also indicate the similarity between the clusters. By cutting the dendrogram at some level, we can obtain a specified number of clusters. By reordering the objects such that the branches of the corresponding dendrogram do not cross, the data set can be arranged with similar objects placed together. Hierarchical clustering algorithms can be further divided into *agglomerative* approaches and *divisive* approaches based on how the hierarchical dendrogram is formed.

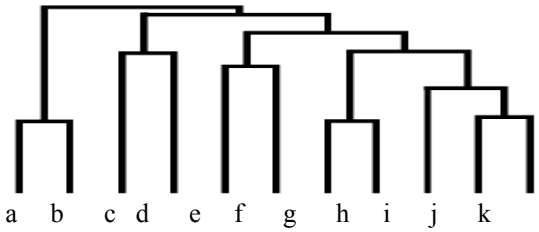


Fig.2.1 Dendogram

2.1.3 Agglomerative Algorithm

All agglomerative techniques naturally form a hirerachical cluster structure in which genes have a crisp membership. Eisen et al. studied GE in the budding yeast, *Saccharmyces cerevisiiae,* using hierarchical methods which have been popularized due to ease of implementation, visualization capability and availability. Methods vary with respect to choice of distance metric, decision on cluster merging, (linkage), as well as parameter selection affecting structure and relationship between clusters. Options include: *single linkage* (cluster separation as distance between two nearest objects), *complete linkage* (as previously, but between two furthest objects), *average linkage* (average distance between all pairs), *centroid* (distance between centroid's of each cluster).

2.1.4 Divisive Clustering

In divisive clustering, all items are initially placed in one cluster and clusters are repeatedly split in two until all items are in their own cluster. The idea is to split clusters where some elements are not sufficiently close other elements.

2.1.5 Linkage Metrics as Proximity Measures between clusters

Let D:=N×N be the dissimilarity matrix between individual objects (also called connectivity matrix), C_1, C_2 two clusters $c_{1i} \in C_1$, $c_{2j} \in C_2$.

$d(C_1, C_2)=\max(d(c_{1i}, c_{2j}) \in D)$

(Complete link)

$d(C_1, C_2)=\min(d(c_{1i}, c_{2j}) \in D)$

(single link)

$d(C_1, C_2)=\frac{1}{|C_1||C_2|}\sum_{i=1}|C_1|\sum_{j=1}|C_2|d(c_{1i}, c_{2j}) \in D$

(Average link)

Other metrics are available: centroid, median, minimum variance, cosine, etc.

2.1.6 Partitional Clustering

Partitive clustering techniques divide data into clusters depending on similarity measures. Widely used methods measure distance from a gene vector to a prototype vector representing the cluster, maximising intra-cluster distance whilst minimizing intercluster distance (e.g. *K-means*, *Fuzzy C-Means*, *SOM*). A major drawback of techniques in this category is that the number of clusters in the data must be specified beforehand, although methods have been developed to try to overcome this.

2.1.7 Graph theoretical Approaches

Given a data set X, we can construct a proximity matrix P, where $P[i, j]$=proximity(O_i, O_j), and a weighted graph G(V,E), called a proximity graph,

where each data pointcorresponds to a vertex. For some clustering methods, each pair of objects is connected by an edge with weight assigned according to the proximity value between the objects, . For other methods, proximity is mapped only to either 0 or 1 on the basis of some threshold, and edges only exist between objects i and j, where P[i,j] equals 1. Graph-theoretical clustering techniques are explicitly presented in terms of a graph, thus converting the problem of clustering a data set into such graph theoretical problems as finding minimum cut or maximal cliques in the proximity graph G.

2.2Sample based Clustering

Within a gene expression matrix, there are usually several particular macroscopic phenotypes of samples related to some diseases or drug effects, such as diseased samples, normal samples, or drug treated samples. The goal of sample-based clustering is to find the phenotype structures or substructures of the samples. The goal of sample-based clustering is to find the phenotype structures of the samples. The clustering techniques can be divided into the following categories and subcategories:

1. clustering based on supervised informative gene selection and
2. unsupervised clustering and informative gene selection
   - unsupervised gene selection and
   - interrelated clustering.

2.3 Subspace based Clustering

For a gene expression matrix containing n genes and m samples, the computational complexity of a complete combination of genes and samples is 2^{n+m} so that the problem of globally optimal block selection is NP-hard. The subspace clustering methods usually define models to describe the target block and then adopt some heuristics to search in the gene-sample space. In the following section, we will discuss some representative subspace clustering algorithms proposed for gene expression matrices.

Gene expression matrices have been extensively analyzed in two dimensions: the gene dimension and the condition dimension. This correspond to the:

- Analysis of expression patterns of genes by comparing rows in the matrix.
- Analysis of expression patterns of samples by comparing columns in the matrix.

Common objectives pursued when analyzing gene expression data include:

1) Grouping of genes according to their expression under multiple conditions.
2) Classification of a new gene, given its expression and the expression of other genes, with known classification.
3) Grouping of conditions based on the expression of a number of genes.
4) Classification of a new sample, given the expression of the genes under that experimental condition.

Clustering techniques can be used to group either genes or conditions, and, therefore, to pursue directly objectives 1 and 3, above, and, indirectly, objectives 2 and 4.

We can then conclude that, unlike clustering algorithms, biclustering algorithms identify groups of genes that show similar activity patterns under a specific subset of the experimental conditions. Therefore, biclustering approaches are the key technique to use when one or more of the following situations applies:

1) Only a small set of the genes participates in a cellular process of interest.
2) An interesting cellular process is active only in a subset of the conditions.
3) A single gene may participate in multiple pathways that may or not be co-active under all conditions.

For these reasons, biclustering algorithms should identify groups of genes and conditions, obeying the following restrictions:

- A cluster of genes should be defined with respect to only a subset of the conditions.
- A cluster of conditions should be defined with respect to only a subset of the genes.
- The clusters should not be exclusive and/or exhaustive: a gene or condition should be able to belong to more than one cluster or to no cluster at all and be grouped using a subset of conditions or genes, respectively.

2.3.1 Definition and Problem Formulation

We will be working with an n by m matrix, where element a_{ij} will be, in general, a given real value.In the case of gene expression matrices, a_{ij} represents the expression level of gene i under condition j. Table I illustrates the arrangement of a gene expression matrix.

TABLE 1

GENE EXPRESSION DATA MATRIX

	Condition 1	...	Condition *j*	...	Condition *m*
Gene 1	a_{11}	...	a_{1j}	...	a_{1m}
Gene ...	...	...	...	...	...
Gene *i*	a_{i1}	...	a_{ij}	...	a_{im}
Gene ...	...	...	...	...	...
Gene *n*	a_{n1}	...	a_{nj}	...	a_{nm}

A large fraction of applications of biclustering algorithms deal with gene expression matrices. However, there are many other applications for biclustering. For this reason, we will consider the general case of a data matrix, *A*, with set of rows *X* and set of columns *Y* , where the elements a_{ij} corresponds to a value representing the relation between row *i* and column *j*.

Such a matrix *A*, with *n* rows and *m* columns, is defined by its set of rows, $X = \{x_1, ..., x_n\}$, and its set of columns, $Y = \{y_i, ..., y_m\}$. We will use (*X,Y*) to denote the matrix *A*. If $I \subseteq X$ and $J \subseteq Y$ are subsets of the rows and columns, respectively, $A_{IJ} = (I, J)$ denotes the sub-matrix A_{IJ} of *A* that contains only the elements a_{ij} belonging to the sub-matrix with set of rows *I* and set of columns *J*.

Given the data matrix *A* a *cluster of rows* is a subset of rows that exhibit similar behavior across the set of all columns. This means that a row cluster A_{IY} = (*I, Y*) is a subset of rows defined over the set of all columns *Y* , where $I = \{i_1, ..., i_k\}$ is a subset of rows $(I \subseteq X$ and $k \leq n)$. A cluster of rows (*I,Y*) can thus be defined as a *k* by *m* sub-matrix of the data matrix *A*. Similarly, a *cluster of columns* is a subset of columns that exhibit similar behavior across the set of all rows. A cluster A_{XJ} = (*X,J*) is a subset of columns defined over the set of all rows *X*, where $J = \{j_1, ..., j_s\}$ is a subset of columns $(J \subseteq Y$ and $s \leq m)$.A cluster of columns (*X,J*) can then be defined as an *n* by *s* sub-matrix of the data matrix *A*.

A *bicluster* is a subset of rows that exhibit similar behavior across a subset of columns, and vice versa. The bicluster A_{IJ} = (*I, J*) is a subset of rows and a subset of columns where $I = \{i_1, ..., i_k\}$ is a subset of rows $(I \subseteq X$ and $k \leq n)$, $J = \{j_1, ..., j_s\}$ and is a subset of columns $(J \subseteq Y$ and $s \leq m)$. A bicluster (*I, J*) can then be defined as a *k* by *s* sub-matrix of the data matrix *A*.

2.3.2 Bicluster Type

An interesting criteria to evaluate a biclustering algorithm concerns the identification of the type of biclusters the algorithm is able to find. We identified four major classes of biclusters:

1) Biclusters with constant values.
2) Biclusters with constant values on rows or columns.
3) Biclusters with coherent values.
4) Biclusters with coherent evolutions.

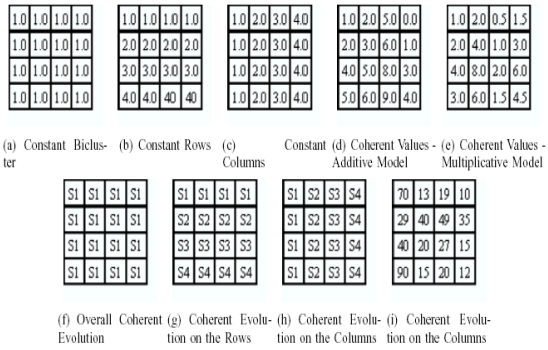


Fig 2.7 : Examples of Different Types of Biclusters

2.3.3 Notation

We will now introduce some notation used in the remaining of the section. Given the data matrix *A* = (*X,Y*), with set of rows *X* and set of columns *Y* , a bicluster is a sub-matrix (*I; J*), where *I* is a subset of the rows *X*, *J* is a subset of the columns *Y* and a_{ij} is the value in the data matrix *A* corresponding to row *i* and column *j*. We denote by a_{ij} the mean of the *i*th row in the bicluster, a_{ij} the mean of the *j*th column in the bicluster and a_{ij} the mean of all elements in the bicluster. These values are defined by:

$$a_{iJ} = \frac{1}{|J|}\sum_{j \in J} a_{ij} \qquad (1)$$

$$a_{Ij} = \frac{1}{|I|}\sum_{i \in I} a_{ij} \qquad (2)$$

$$a_{IJ} = \frac{1}{|I||J|}\sum_{i \in I, j \in J} a_{ij} = \frac{1}{|I|}\sum_{i \in I} a_{iJ} = \frac{1}{|J|}\sum_{j \in J} a_{Ij} \qquad (3)$$

2.3.4 Biclusture Structure

Biclustering algorithms assume one of the following situations: either there is only *one bicluster* in the data matrix (see Fig. (a)), or the data matrix contains *K biclusters*, where *K* is the number of biclusters we

expect to identify and is usually defined *apriori*. While most algorithms assume the existence of several biclusters in the data matrix ,others only aim at finding one bicluster. In fact, even though these algorithms can possibly find more than one bicluster, the target bicluster is usually the one considered the best according to some criterion.

When the biclustering algorithm assumes the existence of several biclusters in the data matrix, the following bicluster structures can be obtained (see Fig. (b) to Fig. (i)):

1) Exclusive row and column biclusters (rectangular diagonal blocks after row and column reorder).
2) Non-Overlapping biclusters with checkerboard structure.
3) Exclusive-rows biclusters.
4) Exclusive-columns biclusters.
5) Non-Overlapping biclusters with tree structure.
6) Non-Overlapping non-exclusive biclusters.
7) Overlapping biclusters with hierarchical structure.
8) Arbitrarily positioned overlapping biclusters.

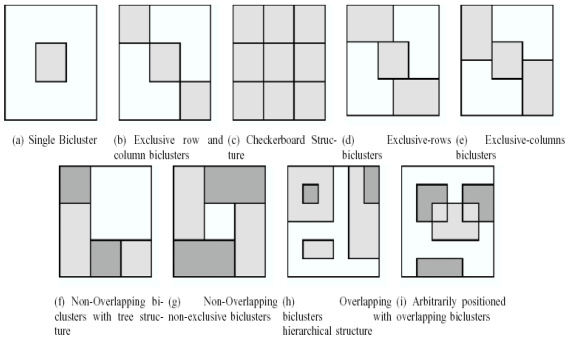


Fig:2.8 Biclusture Structure.

3 BICLUSTERING ALGORITHMS

Biclustering algorithms may have two different objectives: to identify one or to identify a given number of biclusters. Some approaches attempt to identify *one bicluster at a time*. Cheng and Church and Sheng et al., for instance, identify a bicluster at a time, mask it with random numbers, and repeat the procedure in order to eventually find other biclusters. Lazzeroni and Owen also attempt to discover one bicluster at a time in an iterative process where a plaid model is obtained. Ben-Dor et al. also identify one bicluster at a time.

Given the complexity of the problem, a number of different heuristic approaches has been used to address this problem. They can be divided into five classes, studied in the following five subsections:

1) Iterative Row and Column Clustering Combination.
2) Divide and Conquer.
3) Greedy Iterative Search.
4) Exhaustive Bicluster Enumeration.
5) Distribution Parameter Identification.

3 IMPLEMENTATION OF ALGORITHMS USING BicAT and EXPANDER

Microarray technology has become a central tool in biological research, and the identification of gene groups with similar expression patterns represents a key step in the analysis of gene expression data. Traditional clustering algorithms partition an expression matrix into submatrices that extend over the whole set of conditions, giving all conditions equal weight. Several biclustering algorithms have been proposed in the literature, each of which has strengths and weaknesses for the application in different biological scenarios (Madeira and Oliveira, 2004). Although implementations are available for some of the proposed biclustering algorithms, each program may be accompanied by a different user interface and use different input and output formats, which in turn makes the application of several methods a time-consuming task. Desirable is a software tool that offers different biclustering approaches within a common framework—to our best knowledge, such a tool has not been available so far. BicAT tries to fill this gap and provides the following functionality:

- **Data handling:** Tree-structured data handling that allows (i) access to all analysis steps and (ii) data export of biclustering and filtering results
- **Data preprocessing:** Normalization (log2, mean centric) and discretization
- **Clustering:** Five biclustering algorithms and two traditional clustering algorithms
- **Data visualization:** Heatmap and profile visualization of biclusters
- **Postprocessing:** Analysis of gene pair occurrence to derive gene interconnection graphs

4.1 BICLUSTERING METHODS

Selected algorithms Four prominent biclustering methods have been chosen for this comparative study according to three criteria:

(1) to what extent the methods have been used or referenced in the community,
(2) whether their algorithmic strategies are similar and therefore better comparable and
(3) whether an implementation was available or could be easily reconstructed based on the original publications.

The selected algorithms, which all are based on greedy search strategies, are Cheng and Church's algorithm CC (Cheng and Church, 2000); Order Preserving Submatrix Algorithm, OPSM (Ben-Dor et al., 2002); Iterative Signature Algorithm, ISA (Ihmels et al., 2002, 2004) and K-means Clustering.

Reference method (Bimax)

We propose a reference method,namely Bimax, that uses a simple data model reflecting the fundamental idea of biclustering, while allowing to determine all optimal biclusters in reasonable time. This method has the benefit of providing a basis to investigate (1) the usefulness of the biclustering concept in general, independently of interfering effects caused by approximate algorithms, and (2) the effectiveness of more complex scoring schemes and biclustering methods in comparison to a plain approach.

4.1.1Implementation Issues

Five prominent biclustering methods have been chosen for this comparative study according to three criteria: (i) to what extent the methods have been used or referenced in the community, (ii)whether their algorithmic strategies are similar and therefore better comparable, and (iii) whether an implementation was available or could be easily reconstructed based on the original publications.

4.1.1MATRIX VIEW

Original Data **Preprocessed data** **Discretized Data**

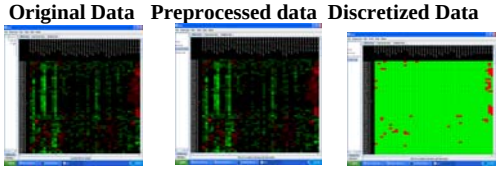


Fig:4.1(a) (b) (c)

BiMax Algorithm

Matrix view **Analysis View** **Expression View**

Fig 4.2

CC Algorithm

Matrix View **Analysis View** **Expression view**

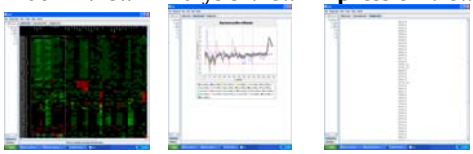


Fig 4.3

ISA Algorithm

Matrix View **Analysis View** **Expression View**

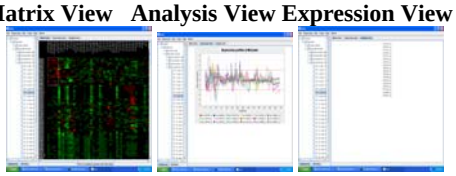


Fig 4.4

OPSM Algorithm

Matrix View **Expression View** **Analysis View**

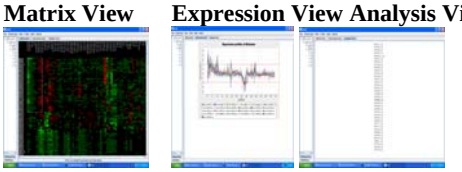
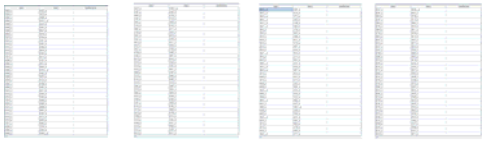


Fig 4.5

4.2Gene Pair Analysis

CC **ISA** **BiMax** **OPSM**



IMPLEMENTATION USING EXPANDER

Biclustering GE Data

Biclustering is clustering of both genes and conditions of the data into subgroups that are not necessarily disjoint

Biclustering is performed by Expander using the SAMBA algorithm In order to apply the SAMBA biclustering algorithm to the data select *Grouping>>Bi-Clustering>>SAMBA*.

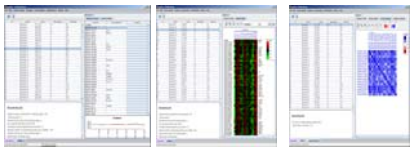


Fig4.7

Matrix Visualizations (Heat maps)

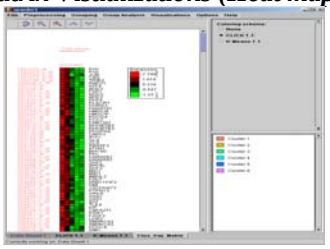


Fig4.8

4.3 COMPARISON METHODOLOGY

First, the comparison focuses on the identification of (locally) co-expressed genes as in in all methods. Classification of samples or inference of regulatory mechanisms may be other tasks for which biclustering can be used; however, considering mainly the gene

dimension has the advantage of various available annotations—in contrast to the condition dimension—and of the possibility to compare the results with classical clustering techniques.

Second, external indices are used to assess the methods under consideration as in most biclustering papers. The reasons are: (1) it is not clear how to extend notions such as homogeneity and separation (Gat-Viks et al., 2003) to the biclustering and (2) there are some issues with internal measures, knowing which Gat-Viks et al. (2003) and Handl et al. (2005) recommend external indices for evaluating the performance of (bi)clustering methods. We consider both synthetic and real datasets for the performance assessment.

Finally, various biclustering concepts and structures can be considered on the basis of which they classify existing biclustering approaches. Here, we investigate two types of bicluster concepts: biclusters with constant expression values and biclusters following an additive model where the expression values are varying over the conditions.

4.4RESULTS

Biological relevance of biclusters with respect to a metabolic pathway map (MPM) for A. thaliana and a protein–protein interaction network (PPI) for S. cerevisiae

TABLE 4

Methods	Proportion of disconnected gene pairs			
	Smaller		Greater	
	MPM	PPI	MPM	PPI
BiMax	58.9	14.0	19.5	64.0
CC	70.0	52.0	15.0	26.0
OPSM	42.8	18.8	28.6	50.0
SAMBA	41.6	0.0	37.5	100.0
ISA	25.0	58.0	25.0	22.0

TABLE 5

Methods	Average shorted distance in the graph			
	Smaller		Greater	
	MPM	PPI	MPM	PPI
BiMax	85.3	58.0	3.4	16.0
CC	70.0	42.0	15.0	34.0
OPSM	92.9	56.3	0.0	43.8
SAMBA	75.6	25.6	13.1	46.2
ISA	50.0	70.0	25.0	22.0

For each bicluster, a Z-test is carried out to check whether its score is significantly smaller or greater than the expected value for random gene groups; the table gives for each method the proportion of biclusters with statistically significant scores (significance level $\alpha = 10^{-3}$). The results for HCL are omitted as all scores equal 0%.

The results for the corresponding comparison for the protein– protein interaction, though, are ambiguous, cf. Table 1. As to the degree of disconnectedness, there is no clear tendency in the data which can be attributed to the fact that not all possible protein pairs have been tested for interaction. Focusing on connected gene pairs only, ISA and Bimax seem to mostly generate gene groups that have a low average distance within the protein network in comparison to random gene sets; for xMotif, the numbers suggest the opposite. Overall, the differences between the biclustering methods demonstrate that special care is necessary when integrating gene expression and protein interaction data: not only the incompleteness of the data needs to be taken into consideration, but also the confidence in the measurements has to be accounted .

5 CONCLUSIONS

The present study compares five prominent biclusterings methods with respect to their capability of identifying groups of (locally) co-expressed genes; hierarchical clustering and a baseline biclustering algorithm, Bimax, proposed in this paper serve as a reference. To this end, different synthetic gene expression data sets corresponding to different notions of biclusters as well as real transcription profiling data are considered. The key results are as follows:

There are significant performance differences among the five biclustering methods. On the real datasets, ISA, Samba and OPSM provide similarly good results: a large portion of the resulting biclusters is functionally enriched and indicates a strong correspondence with known pathways. In the context of the synthetic scenarios, Samba is slightly more robust regarding increased regulatory complexity, but also more sensitive regarding noise than ISA. While Samba and ISA can be used to find multiple biclusters with both constant and coherently increasing values, OPSM is mainly tailored to identify a single bicluster of the latter type. Proposed extensions of the OPSM approach such as Liu and Wang (2003) may resolve these issues.Accordingly, the scores for CC is significantly lower than that for the other biclustering methods under consideration.

The Bimax baseline algorithm presented in this paper achieves similar scores as the best performing biclustering techniques in this study. An advantage of Bimax is that it is capable of generating all optimal biclusters, given the underlying binary data model.

REFERENCES

1] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications," SIGMOD 1998, Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 94-105, 1998.

2] A.A. Alizadeh et al., "Distinct Types of Diffuse Large B-Cell Lymphoma Identified by Gene Expression Profiling," Nature, vol. 403, pp. 503-511, Feb. 2000.

3] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, and A.J. Levine, "Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Array," Proc. Nat'l Academy of Science, vol. 96, no. 12, pp. 6745-6750, June 1999.

[4] S. O. Zakharkin, K. Kim, T. Mehta, L. Chen, S. Barnes, K. E. Scheirer, R. S. Parrish, D. B. Allison, and G. P. Page. Sources of variation in affymetrix microarray experiments. *BMC bioinformatics*, 6:214, Aug 29 2005.

[5] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshi- rani, D. Botstein, and R. B. Altman. Missing value estimation methods for dna microarrays. *Bioinformatics (Oxford, England)*, 17(6):520{525, Jun 2001.

[6] A. G. de Brevern, S. Hazout, and A. Malpertuy. In°uence of microarray experiments missing values on the stability of gene groups by hierarchical clustering. *BMC bioinformatics*, 5:114, Aug 23 2004.

7] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulus, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM/SIGMOD International Conference on Management of Data*, pages 94– 105, 1998.

8] Amir Ben-Dor, Benny Chor, Richard Karp, and Zohar Yakhini. Discovering local structure in gene expression data: The order–preserving submatrix problem. In *Proceedings of the 6th International Conference on Computacional Biology (RECOMB'02)*, pages 49–57, 2002.

9] Pavel Berkhin and Jonathan Becher. Learning simple relations: theory and applications. In *Proceedings of the 2nd SIAM Internationa Conference on Data Mining*, pages 420–436, 2002.

10] Stanislav Busygin, Gerrit Jacobsen, and Ewald Kramer. Double conjugated clustering applied o leukemia microarray data. In *Proceedings of the 2nd SIAM International Conference on Data Mining, Workshop on Clustering High Dimensional Data*, 2002.