Chapter 3

Biclustering

Clustering algorithms have often been used to reduce the complexity of humongous expression data made available with the help of microarray technology. The traditional clustering algorithms are not suitable for all applications especially gene expression data analysis. These traditional clustering algorithms group the genes over all the conditions whereas cellular processes are affected only under a small subset of conditions. Most of the conditions which do not contribute to the cellular process add to the background noise. In gene expression data, it is desired to search for groups of genes which show some compatibility under a small group of conditions. Also, a single gene may belong to more than one group as a gene may be involved in more than one biological process.

Figure 3.1: Principal Component Analysis

Various dimensionality reduction techniques like principal component analysis
(PCA), singular value decomposition and feature selection have been used to filter out the irrelevant conditions. However, these techniques compute clusters on the same set of few relevant conditions (as shown in Figure 3.1) whereas different sets of conditions trigger different biological processes. Biologists are not only interested in identifying sets of coexpressed genes but also the group of conditions responsible for the co-expression of a group of genes. For example, when cancer patients are treated with different drugs, one is interested in determining the genes causing cancer and also the set of drugs to which a patient responds positively.

Cheng and Church [CC00] introduced the notion of biclustering for gene expression data. Biclustering refers to simultaneous clustering of both genes and experimental conditions of the expression data. A bicluster can be viewed as a submatrix of the gene expression matrix such that the rows of the bicluster show a similar behavior under the columns of the bicluster [MO04].

Unlike traditional clustering algorithms where clustering is carried out over all the conditions to produce nonoverlapping clusters, PCA/feature selection algorithms where clustering is performed over the same set of reduced samples or coclustering algorithms which again extract nonoverlapping clusters, biclustering is a more general framework as the biclusters may overlap both on genes as well as on conditions. That is, they allow genes and conditions to belong to more than one bicluster and be responsible for more than one biological activity. Thus biclustering algorithms fit better to biological behavior in contrast to traditional clustering/feature selection/coclustering.

### 3.1 Problem Definition

Let $G$ be a set of $N_g$ genes and $C$ be a set of $N_c$ samples/conditions. Let $E$ be an $N_g \times N_c$ expression matrix where each row represents the expression of a gene under $N_c$ samples. $E$ is subjected to a biclustering algorithm which delivers a biclustering scheme $\pi_i$ con-
sisting of \( k_i \) biclusters. \( \pi_i = (BC_1, BC_2, ..., BC_{k_i}) \), \( BC_j \) is a tuple \((G_j, C_j)\), \( G_j \) being a subset of genes and \( C_j \) a subset of conditions such that \( G_j \) show similar behaviour under \( C_j \). Different biclustering schemes may contain different number of biclusters. Let \( \lambda : (G \times C) \rightarrow 2^{\{0,1,\ldots,k\}} \) be a function that yields a set of labels for each gene condition pair \((g_t, c_r)\). Note that since the biclusters may overlap both on genes and conditions, a (gene, condition) pair may be assigned more than one label. Also, there may be a (gene, condition) pair which does not belong to any bicluster, such a pair is assigned a special label 0.

Many biclustering algorithms also define a score for a bicluster and aim to discover biclusters that optimize the score. Biclustering algorithms are interested in identifying \( B_{opt} = \arg\max\{f(B)\}/\arg\min\{f(B)\} \) where \( f(B) \) denotes the score of a bicluster \( B \). Bicluster scores have been defined in various ways in literature. Cheng and Church used average Mean Square Residue (MSR) as a bicluster score and aimed to minimize the score. The residue of an element \( a_{ij} \) in the bicluster denoted by \( A(I, J) \) is defined as \( r_{ij} = (a_{ij} - a_{IJ} - a_{Ij} - a_{IJ}) \) where \( a_{IJ} \), \( a_{Ij} \) and \( a_{IJ} \) are the row, column and bicluster mean respectively. The mean square residue \( H(I, J) \) of a bicluster \( A(I, J) \) is then given by \( \frac{1}{|I||J|} \sum_{i \in I, j \in J} r_{ij}^2 \). In [LW07] authors define a bicluster score \( s(I, J) \) as the minimum similarity score of any gene with the seed gene \( \min_{i \in I} s(i, J) \) or the minimum similarity score of any condition \( \min_{j \in J} s(I, j) \) whichever is minimum i.e. \( s(I, J) = \min\{\min_{i \in I} \{s(i, J)\}\), \( \min_{j \in J} \{s(I, j)\}\} \) where the similarity is based on Euclidean distance. Liu et al. aimed to maximize the bicluster score. In one of our approaches we use the same definition of the bicluster score but our similarity of a row with the seed row is based on mutual information rather than Euclidean distance.
3.2 Types of Biclusters

Genes in a bicluster have expression values varying in a similar manner or having some relationship under the conditions of the bicluster. Different biclustering algorithms define this similarity/relationship differently. We classify biclusters into two broad categories based on the type of relationship that exists between the genes as follows:

1. **Biclusters with linear relationships**: These biclusters consist of genes having linear relationship between their expression values i.e. the expression levels of genes show linear coherence. Most general form of linear relationship may be described as \( y = mx + c \). Different types of biclusters resulting from different relationships like additive, multiplicative or a combination of both additive and multiplicative as shown in Figure 3.2 respectively fall under this category as shown in Tables (3.1, 3.2 and 3.3). Most of the biclustering algorithms extract such biclusters having linear relationships that may overlap both on genes and conditions.

2. **Biclusters having nonlinear relationships**: Whenever the expression of a gene is a nonlinear function of the expression of another gene we say that there exists a
nonlinear relationship between the two genes as shown in Table 3.4. The expression level of gene $g_2$ is obtained as square of the expression level of gene $g_1$ and that of gene $g_3$ is obtained as absolute of cube of the expression level of gene $g_1$ and that of $g_4$ is obtained as half of square of the expression level of gene $g_1$. Algorithms using similarity measures like distance or correlation coefficient are unable to extract such biclusters.

### 3.3 Organization of Biclusters in the expression data

Different biclustering algorithms extract different number of biclusters of varying sizes. A scheme of biclusters can be classified into **exhaustive** or **nonexhaustive** depending on whether all genes or conditions belong to some bicluster or not as shown in Figure 3.3. A
scheme of biclusters can also be classified into overlapping or nonoverlapping depending upon whether genes or conditions can be a part of one or more than one bicluster at the same time as shown in Figure 3.4.

(a) Exhaustive on both genes and conditions
(b) Exhaustive on genes only
(c) Exhaustive on conditions only
(d) Nonexhaustive biclusters

Figure 3.3: Biclustering schemes with different coverage

1. **Exhaustive on both genes and conditions:** A biclustering scheme is said to be exhaustive on both genes and conditions if all the genes or conditions present in the expression matrix belong to at least one bicluster as shown in Figure 3.3(a).
2. **Exhaustive on genes only**: A biclustering scheme is said to be **exhaustive on genes** if all the genes present in the expression matrix belong to at least one bicluster but there may be certain conditions which are left unclustered as shown in Figure 3.3(b).

3. **Exhaustive on conditions only**: A biclustering scheme is said to be **exhaustive on conditions** if all the conditions present in the expression matrix belong to at least one bicluster but there may be certain genes which are left unclustered as shown in Figure 3.3(c).

4. **Non-Exhaustive**: A biclustering scheme is said to be **non-exhaustive** if few genes or conditions present in the expression matrix are left unclustered as shown in Figure 3.3(d).

5. **Non-overlapping**: Two biclusters are said to be non-overlapping if they neither share a gene nor a condition as shown in Figure 3.4(a).

6. **Overlapping on conditions only**: Two biclusters are said to be **overlapping on conditions** if they share a condition but do not share a gene as shown in Figure 3.4(b). Traditional clusters always overlap on conditions as clustering is performed on the entire set of conditions.

7. **Overlapping on genes only**: When one or more genes may be shared between two biclusters but there are no common conditions then the biclustering scheme is said to be **overlapping on genes** as shown in Figure 3.4(c).

8. **Overlapping on both genes and conditions**: Two biclusters are said to be overlapping on both genes and conditions if they share one or more genes as well as conditions as shown in Figure 3.4(d). Checker board structure allows a gene and condition to belong to more than one bicluster as shown in Figure 3.4(e). However, a gene condition pair can together belong to at most one bicluster.
Figure 3.4: Biclustering schemes with different overlap
9. **Overlapping with hierarchical organization:** When two or more bicluster combine to form another bicluster we get a hierarchical scheme of biclusters as shown in Figure 3.4(f).

### 3.4 Related Work

Like traditional clustering, most of the algorithms for biclustering are also heuristic in nature. Various approaches and similarity measures have been used in literature to extract biclusters. Based on the approach, an algorithm may fall in one or more of the following categories:

1. Iterative Algorithms
2. Enumerative Algorithms
3. Divide and Conquer Algorithms
4. Two way Clustering Algorithms
5. Probabilistic Algorithms
6. Graph based Algorithms
7. Other approaches

#### 3.4.1 Iterative Algorithms

Starting from an initial solution, these algorithms iteratively improve the quality of the biclusters. **Cheng and Church** [CC00] presented the first such algorithm for the biclustering problem. Authors define the residue \( r_{ij} \) of an element \( a_{ij} \) in a bicluster denoted by \( A(I,J) \) as \( r_{ij} = (a_{ij} - a_{Ij} - a_{iJ} - a_{IJ}) \) where \( a_{Ij}, a_{iJ} \) and \( a_{IJ} \) are the row, column and bicluster mean respectively. The mean square residue \( H(I,J) \) is given by

---

31
They defined δ bicluster as the one whose mean square residue score (MSR) is less than a threshold δ. They proposed a node deletion algorithm to find large sized δ biclusters. Starting with the input matrix as a whole the algorithm selects a row or a column with the highest score for deletion such that the MSR of the resultant submatrix is lowered. This is repeated until the MSR is lowered below δ. The bicluster obtained after deletion of rows and columns may not be of maximum size. The algorithm then adds the previously deleted row (column) with the lowest score such that the MSR of the bicluster is increased but remains below δ. This is repeated as long as MSR is less than δ. Missing values in the data are replaced with random numbers. This is in the hope that these random values would not form recognizable pattern and thus would get removed in the node deletion phase.

In order to find more biclusters, the elements of the submatrix representing the discovered bicluster are masked by random numbers. The masking of the discovered bicluster eliminates the related behaviour in it so that other biclusters could be discovered. However, such a masking could interfere with the identification of overlapping biclusters [YWWY03]. The algorithm does not work well when a large amount of noise is present. When the noise levels are high the MSR is also high and some important biclusters may be missed.

**FLexible Overlapped biClusters (FLOC) by Yang et al. [YWWY03]** extended Cheng and Church’s algorithm to find k biclusters simultaneously without random replacement. Starting from a random set of biclusters they iteratively try to move a gene or a condition from one bicluster to another such that the average mean square residue of the entire scheme of biclusters is reduced. FLOC allows biclusters with limited number of missing values. The quality of FLOC’s biclusters depends on the initial seed clusters which are generated randomly.

**Zhang et al. [ZTOT04] in Deterministic Biclustering by Frequent pattern mining (DBF)** extended FLOC by generating the initial seed cluster in a deterministic manner
rather than randomly. They generated a set of good quality (with low mean square residue) biclusters using CHARM [ZH02], a frequent pattern mining algorithm.

**Iterative Signature Algorithm (ISA) by Ihmels et al.** [IFB+02, BIB03] is another iterative algorithm that starts with a random set of genes and computes the set of conditions under which the input genes are most tightly regulated. Using these conditions it iteratively refines the set of genes and then the set of conditions until the set of genes and samples converges i.e. they do not change anymore. ISA works on the hypothesis that although the set of possible input seeds is huge, usually there is only a limited number of fixed points for a given set of thresholds. They run the algorithm for a large number of input seeds and reconstruct the modules from the recurring fixed points by fusing the solutions that were distinct but very similar, using a procedure that resembles agglomerative clustering.

ISA extracts biclusters consisting of genes, which exhibit similar expression pattern with high expression values. The biclusters may be overlapping on both genes and conditions. In presence of high values of expression, ISA misses out biclusters where genes show similar expression pattern but have low values. **Progressive Iterative Signature Algorithm by Kloster et al.** [KTW05] extended ISA to find orthogonal modules. These modules are hard to interpret as they are in different condition space.

Ben Dor et al. [BDCKY02] defined a bicluster as an **Order Preserving Sub Matrix (OPSM)** in which the expression levels of the genes move up and down together on the conditions of the bicluster. For a set $T$ of $s$ conditions, they define a linear order (permutation) $\pi$. A gene belongs to a bicluster defined by $T$ if the $s$ corresponding entries of the gene expression, ordered according to the permutation $\pi$, are strictly increasing. They proposed a heuristic algorithm in which they discover $l$ order preserving submatrix for every possible value for $s$, and select the most significant of these solutions. For a given value of $s$ the algorithm iteratively computes condition permutations starting with a permutation of size 2 (partial linear order), choosing the best $l$ permutations in each
iteration and increasing the number of conditions by one until the permutations are of length $s$. Finally, one best of the $l$ solutions with sample size $s$ are selected. The biclusters retrieved by the algorithm have linear relationships and may overlap both on genes and conditions.

Liu et al. [LW07] projected the biclustering problem as an optimization problem and presented a polynomial time solution for it. With one gene selected as a reference gene, they defined similarity score of a gene as the average Euclidean distance of the gene from the reference gene over a set of conditions. The similarity score of a condition is defined as the average contribution of the sample to the similarity of a set of genes with the reference gene. Finally they defined the similarity score of a bicluster to be the minimum similarity score of rows and columns of the bicluster. The aim is to extract a bicluster with maximum score. Their algorithm is essentially a greedy algorithm which iteratively removes a row or a column that contributes minimum to the score of the bicluster i.e a row or a column whose similarity score is the smallest (worst) among all rows and columns in the current bicluster. Several submatrices are generated in the process; they select the one with the maximum similarity score. More biclusters are generated by changing the reference genes selected randomly by the authors. Instead of generating the gene seeds randomly, a method to select well separated reference genes from the expression data has been used in this work (Chapter 5).

Bozdag et al. [BPC09] proposed an algorithm to extract biclusters named as Correlated Pattern Biclusters (CPB). Pearson’s correlation coefficient (PCC) calculated between any two genes of the CPB over its conditions is required to be greater than a threshold. The algorithm starts by randomly selecting a set of genes and conditions to form a bicluster. It iteratively improves the bicluster by moving genes and conditions in and out of the bicluster. They compare the PCC between each gene and a reference gene to decide which gene to move. To decide on the inclusion of a column $c$ into the bicluster they compute the impact of $c$ on the PCC between the genes of the bicluster. A column $c$
is included only if it does not decrease correlation among the rows in the bicluster.

### 3.4.2 Enumerative Algorithms

Enumerative algorithms extract biclusters from the expression data by listing or enumerating all possible biclusters and then selecting the best amongst them. Wang et al. [WWYY02] used prefix tree to enumerate all the biclusters. They extract \( \delta \) pClusters from the gene expression data (p stands for pattern). A \( \delta \) pCluster is defined as a bicluster in which the change of values on every pair of conditions between every pair of genes is less than a user defined threshold \( \delta \). They try to capture those sets of genes for which the change of expression values on conditions show similar patterns. The algorithm in its first step examines the data to form a set of candidate maximum dimension sets (MDS) for all pairs of genes and for all pairs of conditions. MDSs are pruned using the relationship between the gene pair MDSs and the condition pair MDSs. A prefix tree is then built using the remaining MDSs. Finally the postorder traversal of the tree gives the output biclusters.

Ayadi et al. [AEH09] proposed another enumerative algorithm called BiMine which used a **Binary Enumeration Tree (BET)** to enumerate all biclusters and an evaluation function to throw away the bad quality biclusters. The algorithm proceeds in three steps. The first step involves preprocessing the data during which irrelevant expression values of the data matrix that do not contribute in obtaining relevant results are removed. A gene is considered insignificant if the difference of its expression under every condition and its average under all the conditions is very small. In the next step a BET is constructed from biclusters of single genes and few relevant conditions. The single gene biclusters are then combined to form biclusters of two genes. Two gene biclusters are then combined to construct biclusters of three genes. In this way larger biclusters are built from the smaller ones. At each step the quality of the obtained biclusters is evaluated using an evaluation function based on Spearman’s correlation coefficient. Low quality biclusters
are discarded as there is no point in expanding a bad quality bicluster.

Ahn et al. [AYP11] proposed an algorithm to identify biclusters with functionally highly correlated gene sets called Robust to Noise cluster (RNC). A RNC does not contain genes for which the expression values are constant over a sample pair. A p-RNC is a bicluster where the number of samples are $p$. Initially, the algorithm obtains all the initial 2-RNCs. For all the 2-RNCs, they make a 3-RNC by examining the current sample $s_i$ such that $last < i$ where $s_{last}$ is the last sample in the sample set of 2-RNC. They obtain 3-RNC from 2-RNCs, 4-RNC from 3-RNCs and so on. Those p-RNCs having a larger number of genes and those which have a higher probability of growing to a bigger p-RNC are selected with the help of priority queues. Finally duplicate RNCs are eliminated and the remaining are selected for output.

### 3.4.3 Divide and Conquer Algorithms

These algorithms typically divide the data matrices into a number of submatrices, work recursively on each of these submatrices using some heuristics and select the best biclusters amongst them.

Prelic et al. proposed a fast divide and conquer approach, namely the Binary Inclusion MAXimal biclustering (BIMAX) algorithm [PBZ+06] that finds all inclusion maximal biclusters (that are not entirely contained in any other bicluster). They preprocess the data matrix to convert it into a binary matrix by fixing a threshold. Expression level of gene above the threshold are set to 1 and those below it are set to 0. A bicluster is then defined as a submatrix in which all the elements equal 1. They partition the expression matrix into three submatrices one of which contains only 0 cells and can be neglected. The other two submatrices contain both 0 and 1 cells. The algorithm is recursively applied to these two submatrices. The recursion ends when the reduced matrix represents a bicluster (i.e. contains only 1s). A basic problem with the process of discretization is when the noise levels in the data are high the difference between the bi-
cluster and the background values becomes very small. As a result many small biclusters may be extracted.

### 3.4.4 Two Way Clustering Algorithms

These algorithms view the data matrix in two different ways. The first view considers genes as objects and the conditions as dimensions. The second view considers the conditions as objects and the genes as dimensions. Traditional one way clustering is applied to cluster the genes and conditions separately. The one dimensional clusters are then combined and improved to obtain the final biclusters.

**Coupled Two Way Clustering (CTWC) by Getz et al. [GLD00]** used a one dimensional clustering algorithm called Super Para-magnetic Clustering (SPC) [GLDZ00], to obtain stable (statistically significant) clusters $G_i$ of genes and $C_i$ of conditions. Submatrices are formed by picking one gene cluster from the set $\{G_i\}$ of gene clusters and one condition cluster from the set $\{C_j\}$ of condition clusters. It then recursively computes new biclusters from these submatrices. The process terminates when no new stable biclusters are formed. The type of biclusters obtained depends on the choice of the one way clustering algorithm. SPC uses Euclidean distance as a similarity measure. The algorithm falls in the category of divide and conquer algorithm also.

**Iterative Two Way Clustering (ITWC) by Tang et al. [TZZR01]** used correlation coefficient to perform two way clustering. They identify a reduced set of genes which distinguish the samples from one another. Though Madiera and Oliviera have referred to it as a biclustering algorithm in their survey we feel that the work is closer to the problem of feature selection. Like BiMine, the data is preprocessed to eliminate the genes which do not contribute towards distinguishing the conditions i.e. the genes which do not show much variation on the set of conditions are removed. After preprocessing, the gene clusters are obtained using correlation coefficient. These gene clusters are further used to obtain the condition clusters. The condition clusters are then combined pairwise to form
heterogeneous groups. A reduced set of genes is obtained for each heterogeneous group. Finally using cross validation techniques one reduced gene set is selected for the next iteration. The entire process is repeated iteratively until the number of genes are reduced to some threshold value or the condition clusters reach a certain level of similarity. Finally a set of genes are selected which are used to cluster the conditions. The algorithm also falls in the category of iterative algorithms.

**Double Conjugate Clustering (DCC) by Busygin et al. [BJKA02]** use self organizing maps (SOM) to perform clustering in both the gene space and the condition space. It alternates between clustering genes and conditions. Every node in gene/condition space is assigned a node called the conjugate node in the condition/gene space. In every iteration the nodes of current clustering space are mapped to their conjugate nodes which are moved accordingly. Following this, clustering is done in the other space using SOM. For every gene cluster the corresponding sample cluster contains those samples which can be used to distinguish the genes of the cluster from the rest of the genes. Similarly a gene cluster corresponding to a sample cluster contains those genes which distinguish the samples of the cluster from the rest of the samples. The entire process is repeated iteratively till the number of movements are reduced to a threshold. Finally, a set of gene clusters and its conjugate set of conditions are selected as output. The algorithm also falls under the category of iterative algorithm.

**Chandra et al. [CSM06]** proposed a two way clustering algorithm which use the concept of entropy and correlation coefficient to cluster the genes and fuzzy C-means algorithm to cluster the samples like BiMine and ITWC. Preprocessing of data is done to eliminate the genes that do not contribute to the sample clustering i.e. the genes whose values do not vary much across the samples are eliminated. Next the samples are clustered using Fuzzy C-means algorithm. The number of clusters in the data set are determined by the algorithm itself by measuring the amount of overlap between the clusters. Gene clusters are formed using entropy and the remaining genes are selected for the next iter-
ation resulting in mutually exclusive biclusters. This is repeated till the number of genes are reduced to a minimum limit.

### 3.4.5 Probabilistic Algorithms

Murali and Kasif \[MK03\] defined a bicluster as a set of samples and a set of genes that are conserved under the set of samples. A set of genes is said to be conserved under a set of conditions if it is present in same abundance (state) in all the conditions. A gene state is represented by a range of expression values. They use the term **xMotif** to refer to a bicluster.

For each condition seed $c$, several sets of samples are selected at random. These sets serve as candidates for the discriminating set $D_c$. A discriminating set $D_c$ distinguishes between the genes of the bicluster (those having the same state in all the conditions of $D_c$) and the rest of the genes (those having different values on the conditions of $D_c$). Given a condition seed $c$ and a discriminating set $D_c$, an xMotif contains exactly those genes that have the same state on $c$ and all the conditions in $D_c$. These xMotifs are extracted for all the condition seeds. Finally the one containing the largest number of genes is selected as output. To obtain more biclusters the conditions belonging to the extracted xMotif are removed and the entire process is repeated until all the samples are assigned to some xMotif. It is clear that the biclusters are mutually exclusive and exhaustive on the set of conditions whereas they may overlap and may be non-exhaustive on genes.

### 3.4.6 Graph based Algorithms

**Statistical Algorithmic Method for Bicustering Analysis (SAMBA)** by Tanay et al. \[TSS02\] use graph based techniques along with probabilistic modeling of the data to identify biclusters. They represent the expression data as a bipartite graph whose nodes correspond to genes and conditions. An edge between a gene and a sample represents
significant change in the expression value (up and down regulation) of the gene under that experimental condition with respect to its normal level. Edges and non-edges are assigned weights according to a probabilistic model so that the problem of extracting biclusters is reduced to finding the heavy subgraphs.

Li et al. [LMT+09] proposed a Qualitative BIClustering algorithm (QUBIC) which converts the expression matrix into a representing matrix in which the expression level of a gene under each condition is represented as an integer value. Two genes are considered to be correlated under a subset of conditions if the corresponding integers along the two rows of the representing matrix are identical. They find all optimal submatrices from the representing matrix. For the given matrix, a weighted graph with genes represented as vertices, edges connecting every pair of genes, and the weight of each edge being the similarity level between the two corresponding genes is constructed. The algorithm identifies all biclusters in the matrix by starting with the unused edge as a seed to build an initial bicluster and then iteratively adds more genes till a threshold level is achieved.

3.4.7 Other Algorithms and approaches

Pattern Based Biclustering Algorithms search for specific patterns formed by genes of the bicluster over its conditions. Kluger et al. in [KBCG03] assume that the expression matrix has a checker board like structure. Their method is based on singular value decomposition of the expression matrix. Ben Dor et al. [BDCKY02] also extract biclusters, genes of which show patterns in their expression values. Factor Analysis for Bicluster Acquisition (FABIA) by Hochreiter et al. [HBH+10] is based on a multiplicative model. Two vectors are similar if one is a multiple of the other and the angle between them is zero. The algorithm selects the model parameters using an expectation maximization algorithm.

Gan et al. [GLY08] proposed a geometric interpretation of the biclustering prob-
lem. They show that different types of biclusters are different spatial arrangements of hyperplanes in a high dimensional data space. Thus the biclustering process is reduced to detection of such hyperplanes or linear geometries. They used Hough transform based hyperplane detection algorithm to discover all the hyperplanes that exist in the gene expression data. Each hyperplane is searched for a pattern in the genes lying in it. If a pattern exists it will be selected for output. Tchagang and Tewfix [TT05] proposed Robust Biclustering Algorithm (ROBA) which uses basic linear algebra and arithmetic tools to extract the bicusters. Mitra et al. and others [DMBM07, MB06, MDBM09, NTAR11] use evolutionary approaches to bicluster gene expression data. Conjugate Column Clustering (CCC) by Madiera et al. [MO05] finds biclusters in continuous columns from time series gene expression data.

Other work closely related to biclustering pertains to coclustering [DMM03, ST00] and projected clustering [APW+99, AY00, YCN04]. Though researchers sometimes claim that coclustering, projective clustering and biclustering are all same. But the solutions provided for coclustering and projective clustering do not allow a gene/condition to appear more than once in the biclusters. In [DMM03] Dhillon et al. and in [ST00] Slonim and Tishby have used mutual information for coclustering (simultaneous clustering of rows and columns) word document data. They present the coclustering problem as an optimization problem in which they maximize the mutual information between the clustered random variables subject to restrictions on the number of rows and column clusters. An element $e_{ij}$ of the input matrix represents the frequency of occurrence of $i^{th}$ word in the $j^{th}$ document. Dhillon et al. treat the word document matrix as a co-occurrence matrix and use it to represent the joint probability distribution of the words (represented by random variable $X$) and the documents (represented by random variable $Y$). They approximate the original matrix with a new matrix consisting of a reduced set of rows $\hat{X}$ and a reduced set of columns $\hat{Y}$, so that the new matrix contains as much information about the earlier one as possible. Thus their approach typically leads to dimensionality reduc-
tion. However, it is different from traditional dimensionality reduction in the sense that they do it simultaneously on rows as well as on columns. Though the paper beautifully exploits the information contained in the columns viz a viz rows and the vice versa, it has its limitations especially with reference to gene expression data. Firstly the entries in the gene expression data cannot be treated as a measure of co-occurrence. Secondly, to treat the input matrix as a joint probability distribution the entries must be all positive which may not be the case in gene expression data as down-regulation may be represented by negative values. Banerjee et al. in [BDG+07] propose a generalized coclustering algorithm which works for negative entries in the input matrix as well. They assume that the probability distribution of the input data is either predefined or follows uniform distribution. Both Banerjee and Dhillon identify non-overlapping biclusters whereas a gene may be responsible for more than one cellular function and thus may belong to more than one bicluster. Similarly biclusters may overlap on conditions as well.

3.5 Validation of a Bicluster

A wide variety of clustering and biclustering algorithms exists in literature, yet it is difficult to assess the quality of their solutions. Different algorithms give different solutions on the same data. Most of the time the output depends upon the input parameters as well. Different measures or validity indices are used to evaluate the quality and reliability of the traditional clusters. These measures can be divided into three categories namely internal, external and relative [TK99, HBV01]. Internal measures like intra cluster homogeneity or inter cluster separation rely only on the input data to evaluate the quality of the clusters. External measures use additional information to validate the output. Each cluster can be scored based on prior biological knowledge. For example functional enrichment of the genes in a bicluster can be used to validate the biclusters. Also, quality of a bicluster can be measured by searching for common motifs in the pro-
moter region \([\text{THC}^+99]\) of genes belonging to a bicluster. Relative measures compare the different clustering schemes produced by the same algorithm with different input parameter values. They measure the effect of varying the input parameters on the output of an algorithm.

Different quality measures are applicable in different scenarios depending on the data and on the availability of the ground truth. [GVSS03]. **Rand index** and **Jaccard index** are two measures that are popularly used to assess a clustering solution against the ground truth. Jaccard index is defined as the ratio of the correctly identified objects to the sum of the correctly identified and incorrectly identified objects. Clearly if all the objects are correctly identified the Jaccard index will have the highest value 1 and the least value could be 0. The rand index on the other hand is the ratio of the number of agreements to the number of disagreements. Unlike clusters, different biclusters have different sets of conditions and they may overlap not only on genes but also on conditions. Thus, it is not clear how to extend these measures to biclustering. Also, these measures do not give any indication about the reliability of the biclusters. To the best of our knowledge no general internal index like rand index or jaccard index has been developed for biclustering solutions. Many biclustering algorithms [CC00, YWWY03] have used **mean square residue** (explained earlier) as a measure of quality of biclusters. MSR may be a good measure for distance based approaches and would require normalization of data for it to be meaningful.

### 3.5.1 Biological Validation of Biclusters

Most of the biclustering algorithms use external validation methods like GO annotation term [LW07], metabolic pathways [BIB03], protein protein interaction network [PBZ+06] and patterns in promoter regions [THC+99] to assess the quality of biclusters. These methods are based on the hypothesis that a group of related genes are responsible for some biological activity in a cell. We validated our biclusters using functional annota-
tion (GO terms) and common patterns (motifs) in the promoter regions of the genes of a bicluster with the help of biological tools like DAVID and RSAT as explained ahead.

3.5.2 Functional Annotation using DAVID Toolbox

DAVID (Database for Annotation, Visualization and Integrated Discovery), a free online bioinformatics resource, consisting of knowledge database and analytical tools, that help in extracting biological relevance of a set of genes [DWHL08]. The knowledge database integrates major public bioinformatics resources. DAVID’s knowledge base collects and integrates diverse gene annotation categories, assigns a centralized internal DAVID identifier to each of them in a nonredundant manner. The wide range of biological annotation coverage in the DAVID knowledge base enables a user’s gene ID to be mapped across the entire database thus providing a broad coverage of gene associated annotation. Also, if a significant portion (> 20%) of input gene IDs fail to be mapped to an internal DAVID ID, another DAVID tool, the Gene ID conversion tool starts up to help in the mapping of such IDs.

The Functional Annotation tool of DAVID is used for the enrichment analysis of the gene terms annotated for the input gene set. The basic principle behind the enrichment analysis is that if a biological process is active/abnormal then the co-functioning genes have a higher chance of being selected as a relevant group. To decide about the degree of enrichment, a certain background has to be setup for comparison. As per Huang et al. [DWHL08] larger backgrounds e.g. the total genes in the genome as a background tends to give more significant $p$ values as compared to narrowed down set of genes as background. DAVID has an automatic procedure to determine the background as the global set of genes in the genome on the basis of the user’s uploaded gene list. Thus normally a user does not have to setup a population background by itself. Uploading the gene lists of the bicluster is the first step of analysis. DAVID maps a number of genes in the uploaded list to the associated biological annotation i.e. gene ontology terms using
its functional annotation tool as shown in Figure 3.5. It then statistically examines the enrichment of gene members for each of the annotation terms by comparing the outcome to the reference background. This is done by calculating the $p$ values (defined later) also called as EASE score. Lower is the $p$ value, more statistically significant is the bicluster. Annotation terms below a certain threshold are reported as shown in Figure 3.6.

**Gene Ontology terms**

There are three Gene Ontologies (GO) that form a common language for annotation of genes of different organisms from yeast to human. They relate genes with different biological processes across different species. The three GO ontologies are (i) **Biological process** which include biological functions to which a gene or a gene’s products contribute; (ii) **Cellular component** which includes complex sub-cellular structures, locations and macro-molecular complexes like RNA polymerases where the gene products are active; (iii) **Molecular function** which defines the biochemical activities like carbo-
hydrates binding, ATPase activity etc. of the gene products at the molecular level. A GO term is annotated to a group of genes responsible for a particular biological activity.

\( p \) values

The significance of a bicluster i.e. the likelihood that a bicluster is not found by chance can be measured by statistical measures like \( p \) value. \( p \) values are calculated to measure the statistical significance of functional category enrichment. The GO terms shared by the genes in the user’s list are compared to the background distribution of the annotation. It is the probability of seeing \( x \) or more genes from the input list of \( n \) genes annotated to a particular GO term, given the proportion of genes in the whole genome annotated to that GO Term is \( F \) out of \( G \). Specifically, hyper geometric distribution is used to calculate the probability of observing at least \( x \) or more genes from a functional category from an input gene list of size \( n \) given the background database consists of \( G \) genes out of which \( F \) belong to the functional category.
This is same as calculating the chance of getting atleast \( x \) successes and can also be represented as

\[
p value = 1 - \sum_{j=0}^{x-1} \binom{F_j}{j} \binom{G-F_n-j}{G_n} \tag{3.2}
\]

It is clear that smaller the \( p \) value, more significant is the association of the particular GO term with the group of genes (i.e. it is less likely that the observed annotation of the particular GO term to a group of genes occurs by chance). There may be several GO terms with different \( p \) values associated with an input set of genes belonging to a bicluster. The best \( p \) value for each category may be used to compare the biclusters.

### 3.5.3 Motif analysis using RSA Toolbox

A set of genes showing similar behavior indicates that they are active or expressed together. As explained in Chapter 2, a gene becomes active when a transcription factor (protein responsible for gene regulation) binds to a Transcription Factor Binding Site (TFBS) or motif in the promoter region of the gene. Thus the genes responsible for one biological activity and hence belonging to a bicluster are expected to have shared elements/patterns/motifs. In order to further validate our biclusters we performed motif analysis of the genes of the biclusters using Requence Sequence Analysis Toolbox (RSA T). RSA T consists of many modular tools for sequence retrieval and motif discovery. These tools can either be accessed separately or be connected in a pipeline. Two of these tools are Retrieve Sequence Tool (RST) and Motif Discovery Tool (MDT). Figure 3.7 summarizes the working of RSA T. A set of genes along with the name of the organism is provided as an input to RST as shown in Figure 3.8. RST provides the sequences of the input genes as output which is then fed to MDT to extract the motifs. The output of MDT includes the motifs and their corresponding \( E \) values as shown in Figure 3.9. The \( E \)
value gives the statistical significance of the motif detected. It is the expected number of times a similarity would be observed by chance in a target database of random motifs. It is obtained by multiplying the probability of at least \( n \) occurrences when expecting \( x \) by the number of distinct patterns. Smaller the \( E \) value more significant is the motif detected.

Figure 3.7: Motif analysis using RSAT

Figure 3.8: Snapshot of Retrieve Sequence Analysis Tool
We considered the real datasets used by Prelic et al. [PBZ\textsuperscript{+}06] and by Hochreiter et al. [HBH\textsuperscript{+}10]. \textit{Saccharomyces cerevisiae} also known as brewer’s yeast is a safe, easy to grow, short generation time organism. [Hun93]. As yeasts are eukaryotes and are biochemically similar to humans, they are quite popular with biologists for study purposes. Yeast datasets examines gene expression behaviour during various stress conditions. Expression profiles were normalized (subtracting the mean of each profile and dividing it by the standard deviation across the time points). Another popularly studied organism is \textit{Arabidopsis thaliana}. It is a common weed which undergoes the same processes of growth, development, flowering etc. as most of the higher plants yet has a small genome. It produces a large number of seeds and grows to a mature plant in only about six weeks. We studied the expression dataset of \textit{Saccharomyces cerevisiae}, \textit{Arabidopsis thaliana} and two datasets of homosapiens. The \textit{Human breast cancer} dataset [VDV\textsuperscript{+}02] aimed at pre-
dictive gene signature for the outcome of a breast cancer therapy. The *Diffuse large B-cell lymphoma* dataset [RWC⁺02] contained the gene expression profiles of the lymphomas of patients after chemotherapy. Table 3.5 gives the details about the datasets used.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Genes</th>
<th>Samples</th>
<th>source</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Arabidopsis thaliana</em></td>
<td>619</td>
<td>72</td>
<td><a href="http://www.tik.ee.ethz.ch/sop/bicat">www.tik.ee.ethz.ch/sop/bicat</a></td>
</tr>
<tr>
<td><em>Saccharomyces cerevisiae</em></td>
<td>2993</td>
<td>173</td>
<td><a href="http://www.tik.ee.ethz.ch/sop/bicat">www.tik.ee.ethz.ch/sop/bicat</a></td>
</tr>
<tr>
<td><em>Diffuse large-B-cell lymphoma</em></td>
<td>661</td>
<td>180</td>
<td><a href="http://www.bioinf.jku.at/software/fabia">www.bioinf.jku.at/software/fabia</a></td>
</tr>
<tr>
<td><em>Human breast cancer</em></td>
<td>1213</td>
<td>97</td>
<td><a href="http://www.bioinf.jku.at/software/fabia">www.bioinf.jku.at/software/fabia</a></td>
</tr>
</tbody>
</table>

Table 3.5: Gene Expression Datasets