Cluster Analysis for Gene Expression Data: A Survey

Daxin Jiang   Chun Tang   Aidong Zhang

Department of Computer Science and Engineering

State University of New York at Buffalo

Email: {djiang3, chuntang, azhang}@cse.buffalo.edu

Abstract

DNA microarray technology has now made it possible to simultaneously monitor the expression levels of thousands of genes during important biological processes and across collections of related samples. Elucidating the patterns hidden in gene expression data offers a tremendous opportunity for an enhanced understanding of functional genomics. 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. A very rich literature on cluster analysis has developed over the past three decades. 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.

In this paper, we first briefly introduce the concepts of microarray technology and discuss the basic elements of clustering on gene expression data. In particular, we divide cluster analysis for gene expression data into three categories. Then we present specific challenges pertinent to each clustering category and introduce several representative approaches. We also discuss the problem of cluster validation in three aspects and review various methods to assess the quality

and reliability of clustering results. Finally, we conclude this paper and suggest the promising trends in this field.

Index terms: microarray technology, gene expression data, clustering

1   Introduction

1.1   Introduction to Microarray Technology

1.1.1   Measuring mRNA levels

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* [54] and *oligonucleotide arrays* (abbreviated *oligo chip*) [44]. Despite differences in the details of their experiment protocols, both types of experiments involve three common basic procedures [67]:

- *Chip manufacture:* A microarray is a small chip (made of chemically coated glass, nylon membrane or silicon), onto which tens of thousands of DNA molecules (*probes*) are attached in fixed grids. Each grid cell relates to a DNA sequence.
- *Target preparation, labeling and hybridization:* Typically, two mRNA samples (a test sample and a control sample) are reverse-transcribed into cDNA (*targets*), labeled using either fluorescent dyes or radioactive isotopics, and then hybridized with the probes on the surface of the chip.
- *The scanning process:* Chips are scanned to read the signal intensity that is emitted from the labeled and hybridized targets.

Generally, both cDNA microarray and oligo chip experiments measure the expression level for each DNA sequence by the ratio of signal intensity between the test sample and the control sample, therefore, data sets resulting from both methods share the same biological semantics. In this paper, unless explicitly stated, we will refer to both the cDNA microarray and the oligo chip as *microarray technology* and term the measurements collected via both methods as *gene expression data*.

2

1.1.2 Pre-processing of gene expression data

A microarray experiment typically assesses a large number of DNA sequences (genes, cDNA clones, or expressed sequence tags [ESTs]) under multiple conditions. These conditions may be a time-series during a biological process (e.g., the yeast cell cycle) or a collection of different tissue samples (e.g., normal versus cancerous tissues). 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*** $M = \{w_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq m\}$ (Figure 1(a)), where the rows ($G = \{\vec{g_1}, \ldots, \vec{g_n}\}$) form the expression patterns of genes, the columns ($S = \{\vec{s_1}, \ldots, \vec{s_m}\}$) represent the expression profiles of samples, and each cell w_{ij} is the measured expression level of gene i in sample j. Figure 1 (b) includes some notation that will be used in the following sections.

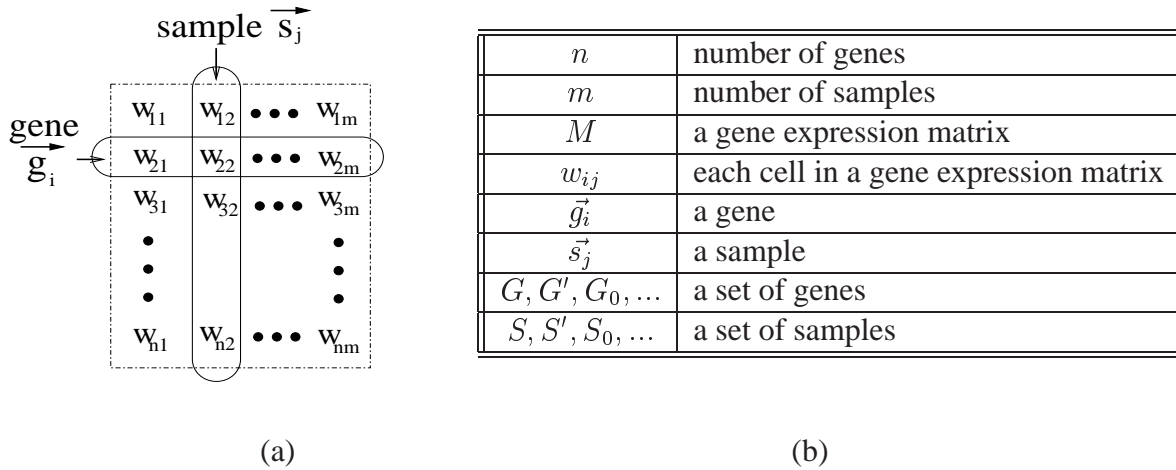| n	number of genes
m	number of samples
M	a gene expression matrix
w_{ij}	each cell in a gene expression matrix
$\vec{g_i}$	a gene
$\vec{s_j}$	a sample
$G, G', G_0, \ldots$	a set of genes
$S, S', S_0, \ldots$	a set of samples

(a)   (b)

Figure 1: (a) A gene expression matrix;   (b) Notation in this paper.

The original gene expression matrix obtained from a scanning process contains noise, missing values, and systematic variations arising from the experimental procedure. Data pre-processing is indispensable before any cluster analysis can be performed. Some problems of data pre-processing have themselves become interesting research topics. Those questions are beyond the scope of this survey; an examination of the problem of *missing value estimation* appears in [69], and the problem of *data normalization* is addressed in [32, 55]. Furthermore, many clustering approaches apply

one or more of the following pre-processing procedures: filtering out genes with expression levels which do not change significantly across samples; performing a logarithmic transformation of each expression level; or standardizing each row of the gene expression matrix with a mean of zero and a variance of one. In the following discussion of clustering algorithms, we will set aside the details of pre-processing procedures and assume that the input data set has already been properly pre-processed.

1.1.3 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 (*co-expressed genes*) can be clustered together with similar cellular functions. This approach may further understanding of the functions of many genes for which information has not been previously available [66, 20]. Furthermore, co-expressed 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 co-regulation. 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 [9, 66]. The inference of regulation through the clustering of gene expression data also gives rise to hypotheses regarding the mechanism of the transcriptional regulatory network [16]. Finally, clustering different samples on the basis of corresponding expression profiles may reveal sub-cell types which are hard to identify by traditional morphology-based approaches [2, 24].

1.2   Introduction to Clustering Techniques

In this subsection, 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.

1.2.1 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 pre-classified objects.

1.2.2 Categories of gene expression data clustering

Currently, a typical microarray experiment contains 10^3 to 10^4 genes, and this number is expected to reach to the order of 10^6. However, the number of samples involved in a microarray experiment is generally less than 100. ***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 [7, 20]. 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. Each group may correspond to some particular macroscopic phenotype, such as clinical syndromes or cancer types [24]. Such ***sample-based clustering*** regards the samples as the objects and the genes as the features. The distinction of gene-based clustering and sample-based clustering is based on different characteristics of clustering tasks for gene expression data. Some clustering algorithms, such as K-means and hierarchical approaches, can be used both to group genes and to partition samples. We will introduce those algorithms as gene-based clustering approaches, and will discuss how to apply them as sample-based clustering in subsection 2.2.1.

Both the gene-based and sample-based clustering approaches search exclusive and exhaustive partitions of objects that share the same feature space. However, current thinking in molecular biology holds that only a small subset of genes participate 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. For subspace clustering algorithms, genes and samples are treated symmetrically, so that either genes or samples can be regarded as objects or features. Furthermore, clusters generated through such algorithms may have different feature spaces.

While a gene expression matrix can be analyzed from different angles, the gene-based, sample-based clustering and subspace clustering analysis face very different challenges. Thus, we may have to adopt very different computational strategies in the three situations. The details of the challenges and the representative clustering techniques pertinent to each clustering category will be discussed in Section 2.

1.2.3   Proximity measurement for gene expression data

Proximity measurement measures the similarity (or distance) between two data objects. Gene expression data objects, no matter genes or samples, can be formalized as numerical vectors $\vec{O}_i = \{o_{ij} | 1 \leq j \leq p\}$, where o_{ij} is the value of the jth feature for the ith data object and p is the number of features. The proximity between two objects O_i and O_j is measured by a *proximity function* of corresponding vectors $\vec{O}_i$ and $\vec{O}_j$.

Euclidean distance is one of the most commonly-used methods to measure the distance between two data objects. The distance between objects O_i and O_j in p-dimensional space is defined as:

$$Euclidean(O_i, O_j) = \sqrt{\sum_{d=1}^{p} (o_{id} - o_{jd})^2}.$$

However, for gene expression data, the overall shapes of gene expression patterns (or profiles) are of greater interest than the individual magnitudes of each feature. Euclidean distance does not score well for shifting or scaled patterns (or profiles) [71]. To address this problem, each object vector is standardized with zero mean and variance one before calculating the distance [66, 59, 56].

An alternate measure is *Pearson's correlation coefficient*, which measures the similarity between the shapes of two expression patterns (profiles). Given two data objects O_i and O_j, Pearson's corre-

lation coefficient is defined as

$$Pearson(O_i, O_j) = \frac{\sum_{d=1}^{p} (o_{id} - \mu_{o_i})(o_{jd} - \mu_{o_j})}{\sqrt{\sum_{d=1}^{p} (o_{id} - \mu_{o_i})^2} \sqrt{\sum_{d=1}^{p} (o_{jd} - \mu_{o_j})^2}},$$

where μ_{o_i} and μ_{o_j} are the means for $\vec{O}_i$ and $\vec{O}_j$, respectively. Pearson's correlation coefficient views each object as a random variable with p observations and measures the similarity between two objects by calculating the linear relationship between the distributions of the two corresponding random variables.

Pearson's correlation coefficient is widely used and has proven effective as a similarity measure for gene expression data [36, 64, 65, 74]. However, empirical study has shown that it is not robust with respect to outliers [30], thus potentially yielding *false positives* which assign a high similarity score to a pair of dissimilar patterns. If two patterns have a common peak or valley at a single feature, the correlation will be dominated by this feature, although the patterns at the remaining features may be completely dissimilar. This observation evoked an improved measure called *Jackknife correlation* [19, 30], defined as $Jackknife(O_i, O_j) = min\{\rho_{ij}^{(1)}, \ldots, \rho_{ij}^{(l)}, \ldots, \rho_{ij}^{(p)}\}$, where $\rho_{ij}^{(l)}$ is the Pearson's correlation coefficient of data objects O_i and O_j with the lth feature deleted. Use of the Jackknife correlation avoids the "dominance effect" of single outliers. More general versions of Jackknife correlation that are robust to more than one outlier can similarly be derived. However, the generalized Jackknife correlation, which would involve the enumeration of different combinations of features to be deleted, would be computationally costly and is rarely used.

Another drawback of Pearson's correlation coefficient is that it assumes an approximate Gaussian distribution of the points and may not be robust for non-Gaussian distributions [14, 16]. To address this, the *Spearman's rank-order correlation coefficient* has been suggested as the similarity measure. The ranking correlation is derived by replacing the numerical expression level o_{id} with its rank r_{id} among all conditions. For example, $r_{id} = 3$ if o_{id} is the third highest value among o_{ik}, where $1 \leq k \leq p$. Spearman's correlation coefficient does not require the assumption of Gaussian distribution and is more robust against outliers than Pearson's correlation coefficient. However, as a consequence of ranking, a significant amount of information present in the data is lost. Our experimental results indicate that, on average, Spearman's rank-order correlation coefficient does not

perform as well as Pearson's correlation coefficient.

Almost all of the clustering algorithms mentioned in this survey use either Euclidean distance or Pearson's correlation coefficient as the proximity measure. When Euclidean distance is selected as proximity measure, the standardization process $\vec{O}'_{id} = \frac{\vec{O}_{id} - \mu_{O_i}}{\sigma_{O_i}}$ is usually applied, where $\vec{O}_{id}$ is the dth feature of object O_i, while μ_{O_i} and σ_{O_i} are the mean and standard deviation of $\vec{O}_i$, respectively. Suppose O'_i and O'_j are the standardized "objects" of O_i and O_j. Then we can prove $Pearson(O_i, O_j) = Pearson(O'_i, O'_j)$ and $Euclidean(O'_i, O'_j) = \sqrt{2p}(\sqrt{1 - Pearson(O'_i, O'_j)})$. These two equations disclose the consistency between Pearson's correlation coefficient and Euclidean distance after data standardization; i.e., if a pair of data objects O_{i1}, O_{j1} has a higher correlation than pair O_{i2}, O_{j2} ($Pearson(O'_{i1}, O'_{j1}) > Pearson(O'_{i2}, O'_{j2})$), then pair O_{i1}, O_{j1} has a smaller distance than pair O_{i2}, O_{j2} ($Euclidean(O'_{i1}, O'_{j1}) < Euclidean(O'_{i2}, O'_{j2})$). Thus, we can expect the effectiveness of a clustering algorithm to be equivalent whether Euclidean distance or Pearson's correlation coefficient is chosen as the proximity measure.

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. In this section, we will introduce the gene-based clustering, sample-based clustering, and subspace clustering techniques, respectively.

2.1   Gene-based Clustering

In this section, we will discuss the problem of clustering genes based on their expression patterns. The purpose of gene-based clustering is to group together co-expressed genes which indicate co-

function and co-regulation. We will first present the challenges of gene-based clustering and then review a series of clustering algorithms which have been applied to group genes. For each clustering algorithm, we will first introduce the basic idea of the clustering process, and then highlight some features of the algorithm.

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. The purpose of clustering gene expression data is to reveal the natural data structures and gain some initial insights regarding data distribution. Therefore, a good clustering algorithm should depend as little as possible on prior knowledge, which is usually not available before cluster analysis. For example, a clustering algorithm which can accurately estimate the "true" number of clusters in the data set would be more favored than one requiring the pre-determined number of clusters.

- Second, due to the complex procedures of microarray experiments, gene expression data often contain a huge amount of noise. Therefore, clustering algorithms for gene expression data should be capable of extracting useful information from a high level of background noise.

- Third, our empirical study has demonstrated that gene expression data are often "highly connected" [37], and clusters may be highly intersected with each other or even embedded one in another [36]. Therefore, algorithms for gene-based clustering should be able to effectively handle this situation.

- 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 (e.g., which clusters are more close to each other, and which clusters are remote from each other), and the relationship between the genes within the same cluster (e.g., which gene can be considered as the representative of the cluster and which genes are at the boundary area of the cluster). A clustering algorithm, which

can not only partition the data set but also provide some graphical representation of the cluster structure would be more favored by the biologists.

2.1.2 K-means

The K-means algorithm [46] is a typical partition-based clustering method. Given a pre-specified number K, the algorithm partitions the data set into K disjoint subsets which optimize the following objective function:

$$E = \sum_{i=1}^{K} \sum_{O \in C_i} |O - \mu_i|^2.$$

Here, O is a data object in cluster C_i and μ_i is the centroid (mean of objects) of C_i. Thus, the objective function E tries to minimize the sum of the squared distances of objects from their cluster centers.

The K-means algorithm is simple and fast. The time complexity of K-means is $O(l * k * n)$, where l is the number of iterations and k is the number of clusters. Our empirical study has shown that the K-means algorithm typically converges in a small number of iterations. However, it also has several drawbacks as a gene-based clustering algorithm. First, the number of gene clusters in a gene expression data set is usually unknown in advance. To detect the optimal number of clusters, users usually run the algorithms repeatedly with different values of k and compare the clustering results. For a large gene expression data set which contains thousands of genes, this extensive parameter fine-tuning process may not be practical. Second, gene expression data typically contain a huge amount of noise; however, the K-means algorithm forces each gene into a cluster, which may cause the algorithm to be sensitive to noise [59, 57].

Recently, several new clustering algorithms [51, 31, 59] have been proposed to overcome the drawbacks of the K-means algorithm. These algorithms typically use some global parameters to control the *quality* of resulting clusters (e.g., the maximal radius of a cluster and/or the minimal distance between clusters). Clustering is the process of extracting all of the qualified clusters from the data set. In this way, the number of clusters can be automatically determined and those data objects which do not belong to any qualified clusters are regarded as outliers. However, the qualities of clusters in gene expression data sets may vary widely. Thus, it is often a difficult problem to

choose the appropriate globally-constraining parameters.

2.1.3 Self-organizing map

The Self-Organizing Map (SOM) was developed by Kohonen [39], on the basis of a single layered neural network. The data objects are presented at the input, and the output neurons are organized with a simple neighborhood structure such as a two dimensional $p * q$ grid. Each neuron of the neural network is associated with a reference vector, and each data point is "mapped" to the neuron with the "closest" reference vector. In the process of running the algorithm, each data object acts as a training sample which directs the movement of the reference vectors towards the denser areas of the input vector space, so that those reference vectors are trained to fit the distributions of the input data set. When the training is complete, clusters are identified by mapping all data points to the output neurons.

One of the remarkable features of SOM is that it generates an intuitively-appealing map of a high-dimensional data set in $2D$ or $3D$ space and places similar clusters near each other. The neuron training process of SOM provides a relatively more robust approach than K-means to the clustering of highly noisy data [62, 29]. However, SOM requires users to input the number of clusters and the grid structure of the neuron map. These two parameters are preserved through the training process; hence, improperly-specified parameters will prevent the recovering of the natural cluster structure. Furthermore, if the data set is abundant with irrelevant data points, such as genes with invariant patterns, SOM will produce an output in which this type of data will populate the vast majority of clusters [29]. In this case, SOM is not effective because most of the interesting patterns may be merged into only one or two clusters and cannot be identified.

2.1.4 Hierarchical clustering

In contrast to partition-based clustering, which attempts to directly decompose the data set into a set of disjoint clusters, 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. Agglomerative algorithms (bottom-up approach) initially regard each data object as an individual cluster, and at each step, merge the closest pair of clusters until all the groups are merged into one cluster. Divisive algorithms (top-down approach) starts with one cluster containing all the data objects, and at each step split a cluster until only singleton clusters of individual objects remain. For agglomerative approaches, different measures of *cluster proximity*, such as single link, complete link and minimum-variance [18, 38], derive various merge strategies. For divisive approaches, the essential problem is to decide how to split clusters at each step. Some are based on heuristic methods such as the deterministic annealing algorithm [3], while many others are based on the graph theoretical methods which we will discuss later.

Eisen et al. [20] applied an agglomerative algorithm called UPGMA (Unweighted Pair Group Method with Arithmetic Mean) and adopted a method to graphically represent the clustered data set. In this method, each cell of the gene expression matrix is colored on the basis of the measured fluorescence ratio, and the rows of the matrix are re-ordered based on the hierarchical dendrogram structure and a consistent node-ordering rule. After clustering, the original gene expression matrix is represented by a colored table (a *cluster image*) where large contiguous patches of color represent groups of genes that share similar expression patterns over multiple conditions.

Alon et al. [3] split the genes through a divisive approach, called the *deterministic-annealing algorithm (DAA)* [53, 52]. First, two initial cluster centroids C_j, $j = 1, 2$, were randomly defined. The expression pattern of gene k was represented by a vector $\vec{g}_k$, and the probability of gene k belonging to cluster j was assigned according to a two-component Gaussian model: $P_j(\vec{g}_k) = exp(-\beta|\vec{g}_k - C_j|^2)/\sum_j exp(-\beta|\vec{g}_k - C_j|^2)$. The cluster centroids were recalculated by $C_j = \sum_k \vec{g}_k P_j(\vec{g}_k)/\sum_k P_j(\vec{g}_k)$. An iterative process (the *EM algorithm*) was then applied to solve P_j and C_j (the details of the EM algorithm will be discussed later). For $\beta = 0$, there was only one cluster, $C_1 = C_2$. When β was increased in small steps until a threshold was reached, two distinct, converged centroids emerged. The whole data set was recursively split until each cluster contained

only one gene.

Hierarchical clustering not only groups together genes with similar expression pattern but also provides a natural way to graphically represent the data set. The graphic representation allows users a thorough inspection of the whole data set and obtain an initial impression of the distribution of data. Eisen's method is much favored by many biologists and has become the most widely-used tool in gene expression data analysis [20, 3, 2, 33, 50]. However, the conventional agglomerative approach suffers from a lack of robustness [62], i.e., a small perturbation of the data set may greatly change the structure of the hierarchical dendrogram. Another drawback of the hierarchical approach is its high computational complexity. To construct a "complete" dendrogam (where each leaf node corresponds to one data object, and the root node corresponds to the whole data set), the clustering process should take $\frac{n^2-n}{2}$ merging (or splitting) steps. The time complexity for a typical agglomerative hierarchical algorithm is $O(n^2 logn)$ [34]. Furthermore, for both agglomerative and divisive approaches, the "greedy" nature of hierarchical clustering prevents the refinement of the previous clustering. If a "bad" decision is made in the initial steps, it can never be corrected in the following steps.

2.1.5 Graph-theoretical approaches

Given a dataset X, we can construct a *proximity matrix P*, where $P[i,j] = proximity(O_i, O_j)$, and a weighted graph $\mathcal{G}(V, E)$, called a *proximity graph*, where each data point corresponds 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 [56, 73]. 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 [7, 26]. Graph-theoretical clustering techniques are explicitly presented in terms of a graph, thus converting the problem of clustering a dataset into such graph theoretical problems as finding minimum cut or maximal cliques in the proximity graph $\mathcal{G}$.

CLICK. *CLICK* (CLuster Identification via Connectivity Kernels) [56] seeks to identify highly connected components in the proximity graph as clusters. CLICK makes the probabilistic assumption that after standardization, pair-wise similarity values between elements (no matter they are in the same cluster or not) are normally distributed. Under this assumption, the weight ω_{ij} of an edge

(i, j) is defined as the probability that vertices i and j are in the same cluster. The clustering process of CLICK iteratively finds the minimum cut in the proximity graph and recursively splits the data set into a set of connected components from the minimum cut. CLICK also takes two post-pruning steps to refine the cluster results. The *adoption step* handles the remaining singletons and updates the current clusters, while the *merging step* iteratively merges two clusters with similarity exceeding a predefined threshold.

In [56], the authors compared the clustering results of CLICK on two public gene expression data sets with those of GENECLUSTER [62] (a SOM approach) and Eisen's hierarchical approach [20], respectively. In both cases, clusters obtained by CLICK demonstrated better quality in terms of homogeneity and separation (these two concepts will be discussed in Section 3). However, CLICK has little guarantee of not going astray and generating highly unbalanced partitions, e.g., a partition that only separates a few outliers from the remaining data objects. Furthermore, in gene expression data, two clusters of co-expressed genes, C_1 and C_2, may be highly intersected with each other. In such situations, C_1 and C_2 are not likely to be split by CLICK, but would be reported as one highly connected component.

CAST. Ben-Dor et al. [7] introduced the idea of a *corrupted clique graph* data model. The input data set is assumed to come from the underlying cluster structure by "contamination" with random errors caused by the complex process of gene expression measurement. Specifically, it is assumed that the true clusters of the data points can be represented by a *clique graph* $\mathcal{H}$, which is a disjoint union of complete sub-graphs with each clique corresponding to a cluster. The similarity graph $\mathcal{G}$ is derived from $\mathcal{H}$ by flipping each edge/non-edge with probability α. Therefore, clustering a dataset is equivalent to identifying the original clique graph $\mathcal{H}$ from the corrupted version $\mathcal{G}$ with as few flips (errors) as possible.

In [7], Ben-Dor et al. presented both a theoretical algorithm and a practical heuristic called *CAST* (Cluster Affinity Search Technique). CAST takes as input a real, symmetric, n-by-n similarity matrix $\mathcal{S}$ ($\mathcal{S}(i, j) \in [0, 1]$ and an *affinity threshold* t. The algorithm searches the clusters one at a time. The currently searched cluster is denoted by C_{open}. Each element x has an *affinity value* $a(x)$ with respect to C_{open} as $a(x) = \sum_{y \in C_{open}} \mathcal{S}(x, y)$. An element x has a high affinity value if it

satisfies $a(x) \geq t|Copen|$; otherwise, x has a low affinity value. CAST alternates between adding high-affinity elements to the current cluster, and removing low-affinity elements from it. When the process stabilizes, C_{open} is considered a complete cluster, and this process continues with each new cluster until all elements have been assigned to a cluster.

The affinity threshold t of the CAST algorithm is actually the average of pairwise similarities within a cluster. CAST specifies the desired cluster quality through t and applies a heuristic searching process to identify qualified clusters one at a time. Therefore, CAST does not depend on a user-defined number of clusters and deals with outliers effectively. Nevertheless, CAST has the usual difficulty of determining a "good" value for the global parameter t.

2.1.6 Model-based clustering

Model-based clustering approaches [21, 76, 23, 45] provide a statistical framework to model the cluster structure of gene expression data. The data set is assumed to come from a finite mixture of underlying probability distributions, with each component corresponding to a different cluster. The goal is to estimate the parameters $\Theta = \{\theta_i \mid 1 \leq i \leq k\}$ and $\Gamma = \{\gamma_r^i \mid 1 \leq i \leq k, 1 \leq r \leq n\}$ that maximize the likelihood $L_{mix}(\Theta, \Gamma) = \sum_{i=1}^{k} \gamma_r^i f_i(x_r|\theta_i)$, where n is the number of data objects, k is the number of components, x_r is a data object (i.e., a gene expression pattern), $f_i(x_r|\theta_i)$ is the density function of x_r of component C_i with some unknown set of parameters θ_i (*model parameters*), and γ_r^i (*hidden parameters*) represents the probability that x_r belongs to C_i. Usually, the parameters Θ and Γ are estimated by the EM algorithm. The EM algorithm iterates between Expectation (E) steps and Maximization (M) steps. In the E step, hidden parameters Γ are conditionally estimated from the data with the current estimated Θ. In the M step, model parameters Θ are estimated so as to maximize the likelihood of complete data given the estimated hidden parameters. When the EM algorithm converges, each data object is assigned to the component (cluster) with the maximum conditional probability.

An important advantage of model-based approaches is that they provide an estimated probability γ_k^i that data object i will belong to cluster k. As we will discuss in Subsection 2.1.1, gene expression data are typically "highly-connected"; there may be instances in which a single gene has a high correlation with two different clusters. Thus, the probabilistic feature of model-based clustering

15

is particularly suitable for gene expression data. However, model-based clustering relies on the assumption that the data set fits a specific distribution. This may not be true in many cases. The modeling of gene expression data sets, in particular, is an ongoing effort by many researchers, and, to the best of our knowledge, there is currently no well-established model to represent gene expression data. Yeung et al. [76] studied several kinds of commonly-used data transformations and assessed the degree to which three gene expression data sets fit the multi-variant Gaussian model assumption. The raw values from all three data sets fit the Gaussian model poorly and there is no uniform rule to indicate which transformation would best improve this fit.

2.1.7   A density-based hierarchical approach: DHC

In [36], the authors proposed a new clustering algorithm, DHC (a *density-based, hierarchical clustering method*), to identify the co-expressed gene groups from gene expression data. DHC is developed based on the notions of "density" and "attraction" of data objects. The basic idea is to consider a cluster as a high-dimensional dense area, where data objects are "attracted" with each other. At the "core" part of the dense area, objects are crowded closely with each other, and thus have high density. Objects at the peripheral area of the cluster are relatively sparsely distributed, and are "attracted" to the "core" part of the dense area.

Once the "density" and "attraction" of data objects are defined, DHC organizes the cluster structure of the data set in two-level hierarchical structures. At the first level, an *attraction tree* is constructed to represent the relationship between the data objects in the dense area. Each node on the attraction tree corresponds to a data object, and the parent of each node is its attractor. The only exception is the data object which has the highest density in the data set. This data object becomes the root of the attraction tree. However, the structure of the attraction tree would be hard to interpret when the data set becomes large and the data structure becomes complicated. To address this problem, at the second structure level, DHC summarizes the cluster structure of the attraction tree into a *density tree*. Each node of the density tree represents a dense area. Initially, the whole data set is considered as a single dense area and is represented by the root node of the density tree. This dense area is then split into several sub-dense areas based on some criteria, where each sub-dense area is represented by a child node of the root node. These sub-dense areas are further split, until

each sub-dense area contains a single cluster.

As a density-based approach, DHC effectively detects the co-expressed genes (which have rel-atively higher density) from noise (which have relatively lower density), and thus is robust in the noisy environment. Furthermore, DHC is particularly suitable for the "high-connectivity" character-istic of gene expression data, because it first captures the "core" part of the cluster and then divides the borders of clusters on the basis of the "attraction" between the data objects. The two-level hi-erarchical representation of the data set not only discloses the relationship between the clusters (via density tree), but also organizes the relationship between data objects within the same cluster (via attraction tree). However, to compute the density of data objects, DHC calculates the distance be-tween each pair of data objects in the data set. The computational complexity of this step is $O(n^2)$, which makes DHC not efficient. Furthermore, two global parameters are used in DHC to control the splitting process of dense areas. Therefore, DHC does not escape from the typical difficulty to determine the appropriate value of parameters.

2.1.8 Summary

In this section, we have reviewed a series of approaches to gene clustering. The purpose of clus-tering genes is to identify groups of highly co-expressed genes from noisy gene expression data. Clusters of co-expressed genes provide a useful basis for further investigation of gene function and gene regulation. Some conventional clustering algorithm, such as K-means, SOM and hierarchical approaches (*UPGMA*), were applied in the early stage and proven to be useful. However, those algorithms were designed for general purpose of clustering, and may not be effective to address the particular challenges for gene-based clustering. Recently, several new clustering algorithms, such as CLICK, CAST and DHC, have been proposed specifically aiming at gene expression data. The experimental study [56, 36] has shown that these new clustering algorithms may provide better performance than the conventional ones on some gene expression data.

However, different clustering algorithms are based on different clustering criteria and/or different assumptions regarding data distribution. The performance of each clustering algorithm may vary greatly with different data sets, and there is no absolute "winner" among the clustering algorithms reviewed in this section. For example, K-means or SOM may outperform other approaches if the

target data set contains few outliers and the number of clusters in the data set is known. While for a very noisy gene expression data set in which the number of clusters is unknown, CAST or CLICK may be a better choice. Table 1 lists some gene expression data sets to which gene-based clustering approaches have commonly been applied.

Data set	Description	Methods
Cho's data [12]	6,220 ORFs in *S. cerevisiae* with 15 time points	K-means [66], SOM [62], CLICK [56], DHC [36]
Iyer's data [33]	9,800 cDNAs with 12 time points	agglomerative hierarchical [20], CLICK [56], DHC [36]
Wen's data [72]	112 rat genes during 9 time points	CAST [7]
Combined yeast data [15, 61, 13]	6,178 ORFs in *S. cerevisiae* during 4 time courses	agglomerative hierarchical [20], model-based [76]
Colon cancer data [3]	6,500 human genes in 40 tumor and 22 normal colon tissue samples	divisive hierarchical [3], model-based [45]
C. elegans data	1246 genes in 146 experiments	CAST [7]
human hematopoietic data	(1) 6000 genes in HL-60 cell lines with 4 time points (2) 6000 genes in four cell lines (HL-60, U937 and Jurkat with 4 time points and NB4 with 5 time points)	SOM [62]

Table 1: Some data sets for gene-based analysis.

2.2 Sample-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. Previous studies [24] have demonstrated that phenotypes of samples can be discriminated through only a small subset of genes whose expression levels strongly correlate with the class distinction. These genes are called *informative genes*. The remaining genes in the gene expression matrix are irrelevant to the division of samples of interest and thus are regarded as noise in the data set.

Although the conventional clustering methods, such as K-means, self-organizing maps (SOM) ,

hierarchical clustering (HC) can be directly applied to cluster samples using all the genes as features, the signal-to-noise ratio (i.e., the number of informative genes versus that of irrelevant genes) is usually smaller than $1 : 10$, which may seriously degrade the quality and reliability of clustering results[73, 63]. Thus, particular methods should be applied to identify informative genes and reduce gene dimensionality for clustering samples to detect their phenotypes.

The existing methods of selecting informative genes to cluster samples fall into two major categories: *supervised analysis* (clustering based on supervised informative gene selection) and *unsupervised analysis* (unsupervised clustering and informative gene selection).

2.2.1 Clustering based on supervised informative gene selection

The supervised approach assumes that phenotype information is attached to the samples, for example, the samples are labeled as diseased vs. normal. Using this information, a "classifier" which only contains the informative genes can be constructed. Based on this "classifier", samples can be clustered to match their phenotypes and labels can be predicted for the future coming samples from the expression profiles. Supervised methods are widely used by biologists to pick up informative genes. The major steps to build the classifier include:

- *Training sample selection.* In this step, a subset of samples is selected to form the training set. Since the number of samples is limited (less than 100), the size of the training set is usually at the same order of magnitude with the original size of samples.

- *Informative gene selection.* The goal of informative gene selection step is to pick out those genes whose expression patterns can distinguish different phenotypes of samples. For example, a gene is uniformly high in one sample class and uniformly low in the other [24]. A series of approaches to select informative genes include: the neighborhood analysis approach [24]; the supervised learning methods such as the support vector machine (SVM) [10], and a variety of ranking based methods [6, 43, 47, 49, 68, 70].

- *Sample clustering and classification.* After about $50 \sim 200$ [24, 42] informative genes which manifest the phenotype partition within the training samples are selected, the whole set of samples are clustered using only the informative genes as features. Since the feature

19

volume is relatively small, conventional clustering algorithms, such as K-means or SOM, are usually applied to cluster samples. The future coming samples can also be classified based on the informative genes, thus the supervised methods can be used to solve sample classification problem.

2.2.2 Unsupervised clustering and informative gene selection

Unsupervised sample-based clustering assumes no phenotype information being assigned to any sample. Since the initial biological identification of sample classes has been slow, typically evolving through years of hypothesis-driven research, automatically discovering samples' phenotypes presents a significant contribution in gene expression data analysis [24]. As an *unsupervised* learning method, clustering also serves as an exploratory task intended to discover unknown sub-structures in the sample space.

Unsupervised sample-based clustering is much more complex than supervised manner since no training set of samples can be utilized as a reference to guide informative gene selection. Many mature statistic methods and other supervised methods can not be applied without the phenotypes of samples known in advance. The following two new *challenges* of unsupervised sample-based clustering make it very hard to detect phenotypes of samples and select informative genes.

- Since the number of samples is very limited while the volume of genes is very large, such data sets are very sparse in high-dimensional genes space. No distinct class structures of samples can be properly detected by the conventional techniques (for example, density based approaches).

- Most of the genes collected may not necessarily be of interest. A small percentage (less than 10% [24]) of genes which manifest meaningful sample phenotype structure are buried in large amount of noise. Uncertainty about which genes are relevant makes it difficult to select informative genes.

Two general strategies have been employed to address the problem of unsupervised clustering and information gene selection: *unsupervised gene selection* and *interrelated clustering*.

20

Unsupervised gene selection.   The first strategy differentiates gene selection and sample clustering as independent processes. First the gene (feature) dimension is reduced, then the conventional clustering algorithms are applied. Since no training samples are available, gene selection only relies on statistical models to analyze the variance in the gene expression data.

Alter et al. [4] applied the principal component analysis (PCA) to capture the majority of the variations within the genes by a small set of principal components (PCs), called "*eigen-genes*." The samples are then projected on the new lower-dimensional PC space. However, eigen-genes do not necessarily have strong correlation with informative genes. Due to the large number of irrelevant genes, discriminatory information of gene expression data is not guaranteed to be the type of user-interested variations. The effectiveness of applying PCA before clustering is discussed in [75].

Ding et al. [17] used a $F-$statistic method to select the genes which show large variance in the expression matrix. Then a min-max cut hierarchical divisive clustering approach is applied to cluster samples. Finally, the samples are ordered such that adjacent samples are similar and samples far away are different. However, this approach relies on the assumption that informative genes exhibit larger variance than irrelevant genes which is not necessarily true for the gene expression data sets [75]. Therefore, the effectiveness of this approach also depends on the data distribution.

Interrelated clustering.   When we have a closer look at the problems of informative gene selection and sample clustering, we will find they are closely interrelated. Once informative genes have been identified, then it is relatively easy to use conventional clustering algorithms to cluster samples. On the other hand, once samples have been correctly partitioned, some supervised methods such as *t-test scores* and *separation scores* [68] can be used to rank the genes according to their relevance to the partition. Genes with high relevance to the partition are considered as informative genes. Based on this observation, the second strategy has been suggested to dynamically use the relationship between the genes and samples and iteratively combine a clustering process and a gene selection process. Intuitively, although we do not know the exact sample partition in advance, for each iteration we can expect to obtain an approximate partition that is close to the target sample partition. The approximate partition allows the selection of a moderately good gene subset, which will, hopefully, draw the approximate partition even closer to the target partition in the next iteration. After several iterations,

the sample partition will converge to the true sample structure, and the selected genes will be feasible candidates for the set of informative genes.

Xing et al. [73] presented a sample-based clustering algorithm named *CLIFF* (CLustering via Iterative Feature Filtering) which iteratively use sample partitions as a reference to filter genes. In [73], non-informative genes were divided into the following three categories: 1) non-discriminative genes (genes in the "off" state); 2) irrelevant genes (genes do not respond to the physiological event); and 3) redundant genes (genes that are redundant or secondary responses to the biological or experimental conditions that distinguish different samples). CLIFF first uses a *two-component Gaussian model* to rank all genes in terms of their discriminability and then select a set of most discriminant genes. It then applies a graph-theoretical clustering algorithm, *NCut* (Approximate Normalized Cut), to generate an initial partition for the samples and enters an iteration process. For each iteration, the input is a reference partition C of the samples and the selected genes. First a scoring method, named *information gain ranking*, is applied to select a set of most "relevant" genes based on the sample partition C. The *Markov blanket filter* is then used to filter "redundant" genes. The remaining genes are used as the features to generate a new partition C' of the samples by NCut clustering algorithm. The new partition C' and the remaining genes will be the input of the next iteration. The iteration ends if this new partition C' is identical to the input reference partition C. However, this approach is sensitive to the outliers and noise of the samples since the gene filtering highly depends on the result of the NCut algorithm which is not robust to the noise and outliers.

Tang et al. [64, 65] proposed iterative strategies for interrelated sample clustering and informative gene selection. The problem of sample-based clustering is formulated via an interplay between sample partition detection and irrelevant gene pruning. The interrelated clustering approaches contained three phases: an *initialization partition* phase, an *interrelated iteration* phase, and a *class validation* phase. In the first phase, samples and genes are grouped into several exclusive smaller groups by conventional clustering methods K-means or SOM. In the iteration phase, the relationship between the groups of the samples and the groups of the genes are measured and analyzed. A *representation degree* measurement is defined to detect the sample groups with high internal coherence as well as large difference between each other. Sample groups with high *representation degree* are posted to form a partial or approximate sample partition called *representative pattern*.

The *representative pattern* is then used to direct the elimination of irrelevant genes. In turn, the remaining meaningful genes were used to guide further *representative pattern* detection. The termination of the series of iterations is determined by evaluating the quality of the sample partition. This is achieved in the class validation phase by assigning *coefficient of variation (CV)* to measure the "internally-similar and well-separated" degree of the selected genes and the related sample partition. The formula for the coefficient of variation is: $CV = \frac{1}{K} \sum_{t=1}^{K} \frac{\sigma_t}{||\vec{\mu_t}||}$, where K represents the number of sample groups, $\vec{\mu_t}$ indicates the center sample vector of group k, and σ_t represents the standard deviation of group t. When a stable and significant sample partition emerges, the iteration stops, and the finial sample partition become the result of the process. This approach delineates the relationships between sample groups and gene groups while conducting an iterative search for samples' phenotypes and informative genes. Since the *representative pattern* identified in each step is only formed by "internally-similar and well-separated" sample groups, this approach is robust to the noise and outliers of the samples.

2.2.3   Summary

In this section, we reviewed a series of approaches to sample-based clustering. 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 sub-categories.

- Clustering based on supervised informative gene selection
- Unsupervised clustering and informative gene selection
    - Unsupervised gene selection
    - Interrelated clustering

Since the percentage of the informative genes is rather low, the major challenge of sample-based clustering is informative gene selection. Supervised informative gene selection techniques which use samples' phenotype information to select informative genes is widely applied, and relatively easy to get high clustering accuracy rate since majority of the samples are used as the training set to select informative genes.

Unsupervised sample-based clustering as well as informative gene selection is complex since no prior knowledge is supposed to be known in advance. Basically, two strategies have been adopted

to address this problem. The first strategy reduces the number of genes before clustering samples. However, since these approaches rely on some statistical models [4, 17], the effectiveness of them heavily depends on the data distribution [75]. Another strategy utilizes the relationship between the genes and samples to perform gene selection and sample clustering simultaneously in an iterative paradigm. Two novel approaches based on the this idea were proposed by Xing et al. [73] and Tang et al. [65]. For both approaches, the iteration converges into an accurate partition of the samples and a set of informative genes as well. One drawback of these approaches is that the gene filtering process is non-invertible. The deterministic filtering will cause data to be grouped based on local decisions.

There are two more issues regarding the quality of sample-based clustering techniques need to be further discussed.

- *The number of clusters K.*   Usually, the number of phenotypes within a gene expression matrix is very small and known in advance. For example, the number of phenotypes is 2 for the well-known leukemia microarray set [24, 58] which often serves as the benchmark for microarray analysis methods. Thus for sample-based analysis, the number of clusters K is always pre-defined, namely, as an input parameter of the clustering method.

- *Time complexity of the sample-based clustering techniques.*   Since the number of samples m is far less than the volume of genes n in a typical gene expression data set, we only investigate the time complexity of the algorithms with respect to the volume of genes n. The time complexity of supervised informative gene selection and unsupervised gene selection approaches is usually $O(n)$ because each gene only needs to be checked once. The time complexity of interrelated clustering methods is $O(n \cdot l)$, while l is the number of iterations which usually is hard to estimate.

Table 2 lists some widely used gene expression data sets for sample-based clustering methods. Since there are many supervised clustering methods, their applications are not exhaustively listed.

2.3   Subspace Clustering

The clustering algorithms discussed in the previous sections are examples of "*global clustering*"; for a given data set to be clustered, the feature space is globally determined and is shared by all

Data set	Description	Methods
Leukemia data [24]	7, 129 genes, 72 samples (25 AML, 47 ALL)	some supervised methods Xing et al.[73], Ding et al.[17]
Lymphoma data [2]	4, 096 genes, 96 samples (46 DLBCL, 50 Nomal)	some supervised methods, Ding et al.[17], Hastie et al.[27]
Colon cancer data [3]	2, 000 genes, 62 samples (22 Normal, 40 Tumor)	some supervised methods
Hereditary breast cancer data [28]	3, 226 genes, 22 samples (7 BRCA1, 8 BRCA2, 7 Sporadics)	some supervised methods
Multiple sclerosis data [48]	4, 132 genes $\times$ 44 samples (15 MS, 14 IFN, 15 Control)	Tang et al.[64, 65]

Table 2: Some data sets for sample-based analysis.

resulting clusters, and the resulting clusters are exclusive and exhaustive. However, it is well known in molecular biology that only a small subset of the genes participates in any cellular process of interest and that any cellular process takes place only in a subset of the samples. Furthermore, a single gene may participate in multiple pathways that may or may not be coactive under all conditions, so that a gene can participate in multiple clusters or in none at all. Recently a series of *subspace clustering* methods have been proposed [22, 11, 40] to capture coherence exhibited by the "*blocks*" within gene expression matrices. In this context, a "*block*" is a sub-matrix defined by a subset of genes on a subset of samples.

Subspace clustering was first proposed by Agrawal et al. in general data mining domain [1] to find subsets of objects such that the objects appear as a cluster in a subspace formed by a subset of the features. Figure 2 shows an example of the subspace clusters (A and B) embedded in a gene expression matrix. In subspace clustering, the subsets of features for various subspace clusters can be different. Two subspace clusters can share some common objects and features, and some objects may not belong to any subspace cluster.

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 subsection, we will discuss some representative subspace clustering algorithms proposed for gene expression matrices. In these representative subspace clustering algorithms, genes and samples are
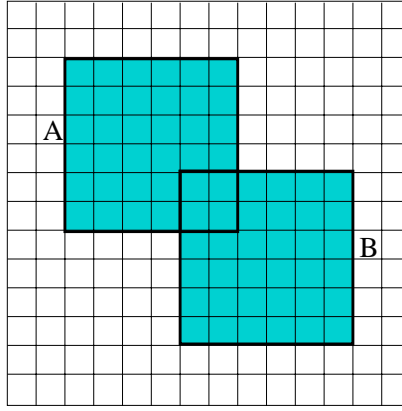
Figure 2: Illustration of subspace clusters.

treated symmetrically such that either genes or samples can be regarded as objects or features.

2.3.1 Coupled two-way clustering (CTWC)

Getz et al. [22] model the block as a stable cluster with features ($\mathcal{F}_i$) and objects ($\mathcal{O}_j$), where both $\mathcal{F}_i$ and $\mathcal{O}_j$ can be either genes or samples. The cluster is "stable" in the sense that, when only the features in $\mathcal{F}_i$ are used to cluster the corresponding $\mathcal{O}_j$, $\mathcal{O}_j$ does not split below some threshold. CTWC provides a heuristic to avoid brute-force enumeration of all possible combinations. Only subsets of genes or samples that are identified as stable clusters in previous iterations are candidates for the next iteration.

CTWC begins with only one pair of gene set and sample set (G_0,S_0), where G_0 is the set containing all genes and S_0 is the set that contains all samples. A hierarchical clustering method, called the *super-paramagnetic clustering algorithm* (SPC) [8], is applied to each set, and the stable clusters of genes and samples yielded by this first iteration are G_1^i and S_1^j. CTWC dynamically maintains two lists of stable clusters (*gene list GL* and *sample list SL*) and a *pair list* of pairs of gene and sample subsets (G_n^i,S_m^j). For each iteration, one gene subset from GL and one sample subset from SL that have not been previously combined are coupled and clustered mutually as objects and features. Newly-generated stable clusters are added to GL and SL, and a pointer that identifies the parent pair is recorded in the pair list to indicate the origin of the clusters. The iteration continues until no new clusters are found which satisfy some criterion, such as stability or critical size.

CTWC was applied to a leukemia data set [24] and a colon cancer data set [3]. For the leukemia

data set, CTWC converges to 49 stable gene clusters and 35 stable sample clusters in two iterations. For the colon cancer data set, 76 stable sample clusters and 97 stable gene clusters were reported by CTWC in two iterations. The experiments demonstrated the capability of CTWC to identify sub-structures of gene expression data which cannot be clearly identified when all genes or samples are used as objects or features.

However, CTWC searches for blocks in a deterministic manner and the clustering results are therefore sensitive to initial clustering settings. For example, suppose (G,S) is a pair of stable clusters. If, during the previous iterations, G was separately assigned to several clusters according to features S', or S was separated in several clusters according to features G', then (G,S) can never be found by CTWC in the following iterations. Another drawback of CTWC is that clustering results are sometimes redundant and hard to interpret. For example, for the colon cancer data, a total of 76 sample clusters and 97 gene clusters were identified. Among these, four different gene clusters partitioned the samples in a normal/cancer classification and were therefore redundant, while many of the clusters were not of interest, i.e., hard to interpret. More satisfactory results would be produced if the framework can provide a systematic mechanism to minimize redundancy and rank the resulting clusters according to significance.

2.3.2   Plaid model

The *plaid model* [40] regards gene expression data as a sum of multiple "*layers*", where each layer may represent the presence of a particular biological process with only a subset of genes and a subset of samples involved. The generalized plaid model is formalized as $Y_{ij} = \sum_{k=0}^{K} \theta_{ijk}\rho_{ik}\kappa_{jk}$, where the expression level Y_{ij} of gene i under sample j is considered coming from multiple sources. To be specific, θ_{ij0} is the background expression level for the whole data set, and θ_{ijk} describes the contribution from layer k. The parameter ρ_{ik} (or κ_{jk}) equals 1 when gene i (or sample j) belongs to layer k, and equals 0 otherwise.

The clustering process searches the layers in the data set one after another, using the *EM algorithm* to estimate the model parameters. Suppose the first $K-1$ layers have been extracted, the Kth layer is identified by minimizing the sum of squared errors $Q = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(Z_{ij} - \theta_{ijK}\rho_{iK}\kappa_{jk}\right)^2$, where $Z_{ij} = Y_{ij} - \theta_{ij0} - \sum_{k=1}^{K-1}\theta_{ijk}\rho_{ij}\kappa_{jk}$ is the residual from the first $K-1$ layers. The cluster-

ing process stops when the variance of expression levels within the current layer is smaller than a threshold.

The plaid model was applied to a yeast gene expression data set combined from several time-series under different cellular processes [40]. Totally, 34 layers were extracted from the data set, among which interesting clusters were found. For example, the second layer was recognized as dominated by genes that produce ribosomal proteins involved in protein synthesis in which mRNA is translated. However, the plaid model is based on the questionable assumption that, if a gene participates in several cellular processes, then its expression level is the sum of the terms involved in the individual processes. Thus, the effectiveness and interpret ability of the discovered layers need further investigation.

2.3.3 Biclustering and δ-Clusters

Cheng et al. [11] introduced the *bicluster* concept to model a block, along with a score called the *mean-squared residue* to measure the coherence of genes and conditions in the block. Let G' and S' be subsets of genes and samples. The pair (G', S') specifies a sub-matrix with the mean-squared residue score $H(G', S') = \frac{1}{|G'||S'|} \sum_{i \in G', j \in S'} (w_{ij} - \eta_{iS'} - \eta_{G'j} + \eta_{G'S'})^2$, where $\eta_{iS'} = \frac{1}{|S'|} \sum_{j \in J} w_{ij}$, $\eta_{G'j} = \frac{1}{|G'|} \sum_{i \in I} w_{ij}$, $\eta_{S'J} = \frac{1}{|G'||S'|} \sum_{i \in G', j \in S'} w_{ij}$ are the row and column means and the means in the submatrix. A submatrix is called a δ-*bicluster* if $H(G', S') \leq \delta$ for some $\delta > 0$. A low mean-squared residue score together with a large variation from the constant suggest a good criterion for identifying a block.

However, the problem of finding a minimum set of biclusters to cover all the elements in a data matrix has been shown to be NP-hard. A greedy method which provides an approximation of the optimal solution and reduces the complexity to polynomial-time has been introduced in [11]. To find a bicluster, the score H is computed for each possible row/column addition/deletion, and the action that decreases H the most is applied. If no action will decrease H or if $H \leq \delta$, a bicluster is returned. However, this algorithm (the brute-force deletion and addition of rows/columns) requires computational time $O((n + m) \cdot mn)$, where n and m are the number of genes and samples, respectively, and it is time-consuming when dealing with a large gene expression data sets. A more efficient algorithm based on multiple row/column addition/deletion (the biclustering algorithm) with

time-complexity $O(mn)$ was also proposed in [11]. After one bi-cluster is identified, the elements in the corresponding sub-matrix are replaced (masked) by random numbers. The bi-clusters are successively extracted from the raw data matrix until a pre-specified number of clusters have been identified. However, the biclustering algorithm also has several drawbacks. First, the algorithm stops when a pre-specified number of clusters have been identified. To cover the majority of elements in the data matrix, the specified number is usually large. However, the biclustering algorithm does not guarantee that the biclusters identified earlier will be of superior quality to those identified later, which adds to the difficulty of the interpretation of the resulting clusters. Second, biclustering "mask" the identified biclusters with random numbers, preventing the identification of overlapping biclusters.

Yang at al. [74] present a subspace clustering method named "δ-clusters" to capture K embedded subspace clusters simultaneously. They use average *residue* across every entry in the sub-matrix to measure the coherence within a submatrix. A heuristic move-based method called FLOC (FLexible Overlapped Clustering) is applied to search K embedded subspace clusters. FLOC starts with K randomly selected sub-matrices as the subspace clusters, then iteratively tries to add/remove each row/column into/out of the subspace clusters to lower the *residue* value, until a local minimum *residue* value is reached. The time complexity of the δ-clusters algorithm is $O((n+m)*n*m*k*l)$, where k is the number of clusters and l is the number of iterations. δ-clusters algorithm also requires the number of the clusters be pre-specified. The advantage of the "δ-clusters" approach is that it is robust to missing values since the *residue* of a submatrix only computed by existing values. "δ-clusters" can also detect overlapping embedded subspace clusters.

2.3.4   Summary

For the subspace clustering techniques applied to gene expression data, a cluster is a "block" formed by a subset of genes and a subset of experimental conditions, where the genes in the "block" illustrate coherent expression patterns under the conditions within the same "block". Different approaches adopt different greedy heuristics to approximate the optimal solution and make the problem tractable. The commonly used data sets are listed in Table 3.

Data set	Description	Methods
Leukemia data [24]	7,129 genes, 72 samples	CTWC [22]
Lymphoma data [2]	4,096 genes, 96 samples	Biclustering [11]
Colon cancer data [3]	2,000 genes, 62 samples	CTWC [22]
Cho's data [12]	6,220 ORFs with 15 time pionts	Biclustering [11], δ-clusters [74]
Combined yeast data [15, 61, 13]	6,178 ORFs during 4 time courses	Plaid Model [40]

Table 3: Some data sets for subspace clustering.

3   Class Validation

The previous sections have reviewed a number of clustering algorithms which partition the data set based on different clustering criteria. For gene expression data, clustering results in groups of co-expressed genes, groups of samples with a common phenotype, or "blocks" of genes and samples involved in specific biological processes. However, different clustering algorithms, or even a single clustering algorithm using different parameters, generally result in different sets of clusters. Therefore, it is important to compare various clustering results and select the one that best fits the "true" data distribution. Cluster validation is the process of assessing the quality and reliability of the cluster sets derived from various clustering processes.

Generally, cluster validity has three aspects. First, the quality of clusters can be measured in terms of *homogeneity* and *separation* on the basis of the definition of a cluster: objects within one cluster are similar to each other, while objects in different clusters are dissimilar with each other. The second aspect relies on a given "ground truth" of the clusters. The "ground truth" could come from domain knowledge, such as known function families of genes, or from other sources such as the clinical diagnosis of normal or cancerous tissues. Cluster validation is based on the agreement between clustering results and the "ground truth". The third aspect of cluster validity focuses on the reliability of the clusters, or the likelihood that the cluster structure is not formed by chance. In this section, we will discuss these three aspects of cluster validation.

3.1   Homogeneity and separation

There are various definitions for the homogeneity of clusters which measures the similarity of data objects in cluster C. For example, $H_1(C) = \frac{\sum_{O_i, O_j \in C, O_i \neq O_j} Similarity(O_i, O_j)}{\|C\| \cdot (\|C\| - 1)}$. This definition repre-

sents the homogeneity of cluster C by the average pairwise object similarity within C. An alternate definition evaluates the homogeneity with respect to the "centroid" of the cluster C, i.e., $H_2(C) = \frac{1}{||C||} \sum_{O_i \in C} Similarity(O_i, \bar{O})$, where $\bar{O}$ is the "centroid" of C. Other definitions, such as the representation of cluster homogeneity via maximum or minimum pairwise or centroid-based similarity within C can also be useful and perform well under certain conditions. Cluster separation is analogously defined from various perspectives to measure the dissimilarity between two clusters C_1, C_2. For example, $S_1(C_1, C_2) = \frac{\sum_{O_i \in C_1, O_j \in C_2} Similarity(O_i, O_j)}{||C_1|| \cdot ||C_2||}$ and $S_2(C_1, C_2) = Similarity(\bar{O}_1, \bar{O}_2)$. Since these definitions of homogeneity and separation are based on the similarity between objects, the quality of C increases with higher homogeneity values within C and lower separation values between C and other clusters. Once we have defined the homogeneity of a cluster and the separation between a pair of clusters, for a given clustering result $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$, we can define the homogeneity and the separation of $\mathcal{C}$. For example, Sharan et al. [56] used definitions of $H_{ave} = \frac{1}{N} \sum_{C_i \in \mathcal{C}} ||C_i|| \cdot H_2(C_i)$ and $S_{ave} = \frac{1}{\sum_{C_i \neq C_j} ||C_i|| \cdot ||C_j||} \sum_{C_i \neq C_j} (||C_i|| \cdot ||C_j||) S_2(C_i, C_j)$ to measure the average homogeniety and separation for the set of clustering results $\mathcal{C}$.

3.2 Agreement with reference partition

If the "ground truth" of the cluster structure of the data set is available, we can test the performance of a clustering process by comparing the clustering results with the "ground truth". Given the clustering results $\mathcal{C} = \{C_1, ..., C_p\}$, we can construct a $n * n$ binary matrix C, where n is the number of data objects, $C_{ij} = 1$ if O_i and O_j belong to the same cluster, and $C_{ij} = 0$ otherwise. Similarly, we can build the binary matrix P for the "ground truth" $\mathcal{P} = \{P_1, ..., P_s\}$. The agreement between $\mathcal{C}$ and $\mathcal{P}$ can be disclosed via the following values:

- n_{11} is the number of object pairs (O_i, O_j), where $C_{ij} = 1$ and $P_{ij} = 1$;
- n_{10} is the number of object pairs (O_i, O_j), where $C_{ij} = 1$ and $P_{ij} = 0$;
- n_{01} is the number of object pairs (O_i, O_j), where $C_{ij} = 0$ and $P_{ij} = 1$;
- n_{00} is the number of object pairs (O_i, O_j), where $C_{ij} = 0$ and $P_{ij} = 0$.

Some commonly used indices [25, 60] have been defined to measure the degree of similarity between $\mathcal{C}$ and $\mathcal{P}$:

$$\text{Rand index: } Rand = \frac{n_{11} + n_{00}}{n_{11} + n_{10} + n_{01} + n_{00}},$$

$$\text{Jaccard coefficient: } JC = \frac{n_{11}}{n_{11} + n_{10} + n_{01}},$$

$$\text{Minkowski measure: } Minkowski = \sqrt{\frac{n_{10} + n_{01}}{n_{11} + n_{01}}}.$$

The *Rand index* and the *Jaccard coefficient* measure the extent of agreement between $\mathcal{C}$ and $\mathcal{P}$, while *Minkowski measure* illustrates the proportion of disagreements to the total number of object pairs (O_i, O_j), where O_i, O_j belong to the same set in $\mathcal{P}$. It should be noted that the *Jaccard coefficient* and the *Minkowski measure* do not (directly) involve the term n_{00}. These two indices may be more effective in gene-based clustering because a majority of pairs of objects tend to be in separate clusters and the term n_{00} would dominate the other three terms in both good and bad solutions. Other methods are also available to measure the correlation between the clustering results and the "ground truth" [25]. Again, the optimal index selection is application-dependent.

3.3 Reliability of clusters

While a validation index can be used to compare different clustering results, this comparison will not reveal the reliability of the resulting clusters; that is, the probability that the clusters are not formed by chance. In the following subsection, we will review two approaches to measuring the significance of the derived clusters.

P**-value of a cluster**   In [66], Tavazoie et al. mapped the genes in each resulting cluster to the 199 functional categories in the Martinsried Institute of Protein Sciences function classification scheme (MIPS) database. For each cluster, P-values were calculated to measure the statistical significance for functional category enrichment. To be specific, the authors used the hyper-geometric distribution to calculate the probability of observing at least k genes from a functional category within a cluster

of size n:

$$P = 1 - \sum_{i=0}^{k-1} \frac{\begin{pmatrix} f \\ i \end{pmatrix} \begin{pmatrix} g - f \\ n - i \end{pmatrix}}{\begin{pmatrix} g \\ n \end{pmatrix}},$$

where f is the total number of genes within a functional category and g is the total number of genes within the genome. Since the expectation of P within the cluster would be higher than 0.05%, the authors regarded clusters with P-values smaller than $3 * 10^{-4}$ as significant. Jakt et al. [35] integrated the assessment of the potential functional significance of both gene clusters and the corresponding postulated regulatory motifs (common DNA sequence patterns), and developed a method to estimate the probability (P-value) of finding a certain number of matches to a motif in all of the gene clusters. A smaller probability indicates a higher significance of the clustering results.

Prediction strength   A novel approach to evaluating the reliability of sample clusters is based on the concept that if a clustering result reflects true cluster structure, then a predictor based on the resulting clusters should accurately estimate the cluster labels for new test samples. For gene expression data, extra data objects are rarely used as test samples, since the number of available samples is limited. Rather, a *cross-validation* method is applied. The generated clusters are assessed by repeatedly measuring the *prediction strength* with one or a few of the data objects left out in turn as "test samples" while the remaining data objects are used for clustering.

Golub et al. introduced a method based on this idea. Suppose a clustering algorithm partitions the samples into two groups C_1 and C_2, with samples $\vec{s}_1, \ldots, \vec{s}_k$ belonging to C_1 and samples $\vec{s}_{k+1}, \ldots, \vec{s}_{p-1}$ belonging to C_2, $\vec{s}_p$ being the left out test sample, and G being a set of genes most correlated with the current partition. Given the test sample $\vec{s}_p$, each gene $\vec{g}_i \in G$ votes for either C_1 or C_2, depending on whether the gene expression level w_{ip} of $\vec{g}_i$ in $\vec{s}_p$ is closer to μ_1 or μ_2 (which denote, respectively, the mean expression levels of $\vec{s}_1, \ldots, \vec{s}_k$ (C_1) and $\vec{s}_{k+1}, \ldots, \vec{s}_{p-1}$ (C_2)). The votes for C_1 and C_2 are summed as V_1 and V_2, respectively, and the *prediction strength* for $\vec{s}_p$ is defined as $\left| \frac{V_1 - V_2}{V_1 + V_2} \right|$. Clearly, if most of the genes in G uniformly vote $\vec{s}_p$ for C_1 (or for C_2), the value of the predication strength will be high. High values of prediction strength with respect to sufficient

test samples indicate a biologically significant clustering.

Golub's method constructs a predictor based on derived clusters and converts the reliability assessment of sample clusters to a "supervised" classification problem. In [77], Yeung et al. extended the idea of "prediction strength" and proposed an approach to cluster validation for gene clusters. Intuitively, if a cluster of genes has possible biological significance, then the expression levels of the genes within that cluster should also be similar to each other in "test" samples that were not used to form the cluster. Yeung et al. proposed a specific *figure of merit (FOM)*, to estimate the predictive power of a clustering algorithm. Suppose $C_1, \ldots, C_k$ are the resulting clusters based on samples $1, \ldots, (e-1), (e+1), \ldots, m$, and sample e is left out to test the prediction strength. Let $R(g, e)$ be the expression level of gene g under sample e in the raw data matrix. Let $\mu_{C_i}(e)$ be the average expression level in sample e of the genes in cluster C_i. The *figure of merit* with respect to e and the number of clusters k is defined as

$$FOM(e, k) = \sqrt{\frac{1}{n} * \sum_{i=1}^{k} \sum_{x \in C_i} (R(x, e) - \mu_{C_i}(e))^2}.$$

Each of the m samples can be left out in turn, and the *aggregate figure of merit* is defined as $FOM(k) = \sum_{e=1}^{m} FOM(e, k)$. The FOM measures the mean deviation of the expression levels of genes in e relative to their corresponding cluster means. Thus, a small value of FOM indicates a strong prediction strength, and therefore a high level reliability of the resulting clusters. Levine et al. [41] proposed another figure of merit $\mathcal{M}$ based on a resampling scheme. The basic idea is that the cluster structure derived from the whole data set should be able to "predict" the cluster structure of subsets of the full data. $\mathcal{M}$ measures the extent to which the clustering assignments obtained from the resamples (subsets of the full data) agree with those from the full data. A high value of $\mathcal{M}$ against a wide range of resampling indicates a reliable clustering result.

4 Current and Future Research Directions

Recent DNA microarray technologies have made it possible to monitor transcription levels of tens of thousands of genes in parallel. Gene expression data generated by microarray experiments offer

tremendous potential for advances in molecular biology and functional genomics. In this paper, we reviewed both classical and recently developed clustering algorithms, which have been applied to gene expression data, with promising results.

Gene expression data can be clustered on both genes and samples. As a result, the clustering algorithms can be divided into three categories: gene-based clustering, sample-based clustering and subspace clustering. Each category has specific applications and present specific challenges for the clustering task. For each category, we have analyzed its particular problems and reviewed several representative algorithms.

Given the variety of available clustering algorithms, one of the problems faced by biologists is the selection of the algorithm most appropriate to a given gene expression data set. However, there is no single "best" algorithm which is the "winner" in every aspect. Researchers typically select a few candidate algorithms and compare the clustering results. Nevertheless, we have shown that there are three aspects of cluster validation, and for each aspect, various approaches can be used to assess the quality or reliability of the clustering results. In this instance, as well, there are no existing standard validity metrics. In fact, the performance of different clustering algorithms and different validation approaches is strongly dependent on both data distribution and application requirements. The choice of the clustering algorithm and validity metric is often guided by a combination of evaluation criteria and the user's experience.

A gene expression data set typically contains thousands of genes. However, biologists often have different requirements on cluster granularity for different subsets of genes. For some purpose, biologists may be particularly interested in some specific subsets of genes and prefer small and tight clusters. While for other genes, people may only need a coarse overview of the data structure. However, most of the existing clustering algorithms only provide a crisp set of clusters and may not be flexible to different requirements for cluster granularity on a single data set. For gene expression data, it would be more appropriate to avoid the direct partition of the data set and instead provide a scalable graphical representation of the data structure, leaving the partition problem to the users. Several existing approaches, such as hierarchical clustering, SOM and Optics [5], can graphically represent the cluster structure. However, these algorithms may not be able to adapt to different user requirements on cluster granularity for different subsets of the data.

Clustering is generally recognized as an "unsupervised" learning problem. Prior to undertaking a clustering task, *"global"* information regarding the data set, such as the total number of clusters and the complete data distribution in the object space, is usually unknown. However, some *"partial"* knowledge is often available regarding a gene expression data set. For example, the functions of some genes have been studied in the literature, which can provide guidance to the clustering. Furthermore, some groups of the experimental conditions are known to be strongly correlated, and the *differences* among the cluster structures under these different groups may be of particular interest. If a clustering algorithm could integrate such partial knowledge as some *clustering constraints* when carrying out the clustering task, we can expect the clustering results would be more biologically meaningful. In this way, clustering could cease to be a "pure" unsupervised process and become an interactive exploration of the data set.

References

[1] Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. Automatic subspace clustering of high dimensional data for data mining applications. In *SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data*, pages 94–105, 1998.

[2] Alizadeh, A.A., Eisen, M.B., Davis, R.E., Ma, C., Lossos, I.S., Rosenwald, A., Boldrick, J.C., Sabet, H., Tran, T., Yu, X., Powell, J.I., Yang, L., Marti, G.E., Moore, T., Hudson, J. Jr, Lu, L., Lewis, D.B., Tibshirani, R., Sherlock, G., Chan, W.C., Greiner, T.C., Weisenburger, D.D., Armitage, J.O., Warnke, R., Staudt, L.M. *et al.* Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, Vol.403:503–511, February 2000.

[3] Alon U., Barkai N., Notterman D.A., Gish K., Ybarra S., Mack D. and Levine A.J. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide array. *Proc. Natl. Acad. Sci. USA*, Vol. 96(12):6745–6750, June 1999.

[4] Alter O., Brown P.O. and Bostein D. Singular value decomposition for genome-wide expression data processing and modeling. *Proc. Natl. Acad. Sci. USA*, Vol. 97(18):10101–10106, Auguest 2000.

[5] Ankerst, Mihael, Breunig, Markus M., Kriegel, Hans-Peter, Sander, Jörg. OPTICS: Ordering Points To Identify the Clustering Structure. *Sigmod*, pages 49–60, 1999.

[6] Ben-Dor A., Friedman N. and Yakhini Z. Class discovery in gene expression data. In *Proc. Fifth Annual Inter. Conf. on Computational Molecular Biology (RECOMB 2001)*, pages 31–38. ACM Press, 2001.

[7] Ben-Dor A., Shamir R. and Yakhini Z. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3/4):281–297, 1999.

[8] Blat, M., S. Wiseman and E. Domany. Super-paramagnetic clustering of data. *Phys. Rev. Letters*, 76:3251–3255, 1996.

[9] Brazma, Alvis and Vilo, Jaak. Minireview: Gene expression data analysis. *Federation of European Biochemical societies*, 480:17–24, June 2000.

[10] Brown M.P.S., Grundy W.N., Lin D., Cristianini N., Sugnet C.W., Furey T.S., Ares M.Jr. and Haussler D. Knowledge-based analysis of microarray gene expression data using support vector machines. *Proc. Natl. Acad. Sci.*, 97(1):262–267, January 2000.

[11] Cheng Y., Church GM. Biclustering of expression data. *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB)*, 8:93–103, 2000.

[12] Cho, R. J., Campbell, M. J., Winzeler, E. A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T. G., Gabrielian, A. E., Landsman, D., Lockhart, D. J., and Davis, R. W. A Genome-Wide Transcriptional Analysis of the Mitotic Cell Cycle. *Molecular Cell*, Vol. 2(1):65–73, July 1998.

[13] Chu S., DeRisi J., Eisen M., Mulholland J., Botstein D., Brown PO., et al. The transcriptional program of sporulation in budding yeast. *Science*, 282(5389):699–705, 1998.

[14] David R. Bickel. Robust Cluster Analysis of DNA Microarray Data: An Application of Nonparametric Correlation Dissimilarity. *Proceedings of the Joint Statistical Meetings of the American Statistical Association (Biometrics Section)*, 2001.

[15] DeRisi, J.L. , Iyer, V.R., and Brown, P.O. Exploring the Metabolic and Genetic Control of Gene Expression on a Genomic Scale. *Science*, pages 680–686, 1997.

[16] D'haeseleer, P., Wen, X., Fuhrman, S., Somogyi, R. Mining the Gene Expression Matrix: Inferring Gene Relationships From Large Scale Gene Expression Data. *Information Processing in Cells and Tissues*, pages 203–212, 1998.

[17] Ding, Chris. Analysis of gene expression profiles: class discovery and leaf ordering. In *Proc. of International Conference on Computational Molecular Biology (RECOMB)*, pages 127–136, Washington, DC., April 2002.

[18] Dubes, R. and Jain, A. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[19] Efron, B. The Jackknife, the Bootstrap, and Other Resampling Plans. *CBMS-NSF Regional Conference Series in Applied Mathematics*, 38, 1982.

[20] Eisen, Michael B., Spellman, Paul T., Brown, Patrick O. and Botstein, David . Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, 95(25):14863–14868, December 1998.

[21] Fraley C. and Raftery A.E. How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis. *The Computer Journal*, 41(8):578–588, 1998.

[22] Getz G., Levine E. and Domany E. Coupled two-way clustering analysis of gene microarray data. *Proc. Natl. Acad. Sci. USA*, Vol. 97(22):12079–12084, October 2000.

[23] Ghosh, D. and Chinnaiyan, A.M. Mixture modelling of gene expression data from microarray experiments. *Bioinformatics*, 18:275–286, 2002.

[24] Golub T.R., Slonim D.K., Tamayo P., Huard C., Gassenbeek M., Mesirov J.P., Coller H., Loh M.L., Downing J.R., Caligiuri M.A., Bloomfield D.D., and Lander E.S. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, Vol. 286(15):531–537, October 1999.

[25] Halkidi, M., Batistakis, Y. and Vazirgiannis, M. On Clustering Validation Techniques. *Intelligent Information Systems Journal*, 2001.

[26] Hartuv, Erez and Shamir, Ron. A clustering algorithm based on graph connectivity. *Information Processing Letters*, 76(4–6):175–181, 2000.

[27] Hastie T., Tibshirani R., Boststein D. and Brown P. Supervised harvesting of expression trees. *Genome Biology*, Vol. 2(1):0003.1–0003.12, January 2001.

[28] Hedenfalk, I., Duggan, D., Chen, Y. D., Radmacher, M., Bittner, M., Simon, R., Meltzer, P., Gusterson, B., Esteller, M., Kallioniemi, O. P., Wilfond, B., Borg, A., and Trent, J. Gene-expression profiles in hereditary breast cancer. *The New England Journal of Medicine*, 344(8):539–548, February 2001.

[29] Herrero J., Valencia A. and Dopazo J. A hierarchical unsupervised growing neural network for clustering gene expression patterns. *Bioinformatics*, 17:126–136, 2001.

[30] Heyer, L.J., Kruglyak, S. and Yooseph S. Exploring Expression Data: Identification and Analysis of Coexpressed Genes. *Genome Res*, 1999.

[31] Heyer LJ., Kruglyak S., Yooseph S. Exploring expression data: identification and analysis of coexpressed genes. *Genome Res*, 9(11):1106–1115, 1999.

[32] Hill, A., Brown, E., Whitley, M., Tucker-Kellogg, G., Hunter, C., Slonim D. Evaluation of normalization procedures for oligonucleotide array data based on spiked cRNA controls. *Genome Biology*, 2(12):research0055.1–0055.13, 2001.

[33] Iyer V.R., Eisen M.B., Ross D.T., Schuler G., Moore T., Lee J.C.F., Trent J.M., Staudt L.M., Hudson Jr. J., Boguski M.S., Lashkari D., Shalon D., Botstein D. and Brown P.O. The transcriptional program in the response of human fibroblasts to serum. *Science*, 283:83–87, 1999.

[34] Jain, A.K., Murty, M.N. and Flynn, P.J. Data clustering: a review. *ACM Computing Surveys*, 31(3):254–323, September 1999.

[35] Jakt, L.M., Cao, L., Cheah, K.S.E., Smith, D.K. Assessing clusters and motifs from gene expression data. *Genome research*, 11:112–123, 2001.

[36] Jiang, D., Pei, J. and Zhang, A. . DHC: A Density-based Hierarchical Clustering Method for Time-series Gene Expression Data. In *Proceeding of BIBE2003: 3rd IEEE International Symposium on Bioinformatics and Bioengineering*, Bethesda, Maryland, March 10-12 2003.

[37] Jiang, D., Pei, J. and Zhang, A. . Interactive Exploration of Coherent Patterns in Time-Series Gene Expression Data. In *In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'03)*, Washington, DC, USA, August 24-27 2003.

[38] Kaufman, L. and Rousseeuw, P.J. *Finding Groups in Data: an Introduction to Cluster Analysis*. John Wiley and Sons, 1990.

[39] Kohonen T. *Self-Organization and Associative Memory*. Spring-Verlag, Berlin, 1984.

[40] Lazzeroni, L. and Owen A. Plaid models for gene expression data. *Statistica Sinica*, 12(1):61–86, 2002.

[41] Levine, E., Domany, E. Resampling methods for unsupervised estimation of cluster validity. *Neural computation*, 13:2573–2593, 2001.

[42] Li, Leping, Weinberg, Clarice R., Darden, Thomas A., and Pedersen, Lee G. Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the ga/knn method. *Bioinformatics*, 17:1131–1142, 2001.

[43] Li, Wentian. Zipf's Law in Importance of Genes for Cancer Classification Using Microarray Data. Lab of Statistical Genetics, Rockefeller University, April 2001.

[44] Lockhart, D. *et al.* Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nat. Biotechnol*, 14:1675–1680, 1996.

[45] McLachlan, G.J., Bean R.W. and Peel D. A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics*, 18:413–422, 2002.

[46] McQueen J.B. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, Univ.of California, Berkeley, 1967. Univ.of California Press, Berkeley.

[47] Moler E.J., Chow M.L. and Mian I.S. Analysis of Molecular Profile Data Using Generative and Discriminative Methods. *Physiological Genomics*, Vol. 4(2):109–126, 2000.

[48] Nguyen LT., Ramanathan M., Munschauer F., Brownscheidle C., Krantz S., Umhauer M., et al. Flow cytometric analysis of in vitro proinflammatory cytokine secretion in peripheral blood from multiple sclerosis patients. *J Clin Immunol*, 19(3):179–185, 1999.

[49] Park P.J., Pagano M., and Bonetti M. A Nonparametric Scoring Algorithm for Identifying Informative Genes from Microarray Data. In *Pacific Symposium on Biocomputing*, pages 52–63, 2001.

[50] Perou C.M., Jeffrey S.S., Rijn, M.V.D., Rees C.A., Eisen M.B., Ross D.T., Pergamenschikov A., Williams C.F., Zhu S.X., Lee, J.C.F., Lashkari D., Shalon D., Brown P.O., and Bostein D. Distinctive gene expression patterns in human mammary epithelial cells and breast cancers. *Proc. Natl. Acad. Sci. USA*, Vol. 96(16):9212–9217, August 1999.

[51] Ralf-Herwig, Poustka, A., Muller, C., Bull, C., Lehrach, H. and O'Brien, J. Large-Scale Clustering of cDNA-Fingerprinting Data. *Genome Research*, 9:1093–1105, 1999.

[52] Rose, K. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proc. IEEE*, 96:2210–2239, 1998.

[53] Rose, K., Gurewitz, E. and Fox, G. *Phys. Rev. Lett.*, 65:945–948, 1990.

[54] Schena,M., D. Shalon, R. Davis, and P.Brown. Quantitative monitoring of gene expression patterns with a compolementatry DNA microarray. *Science*, 270:467–470, 1995.

[55] Schuchhardt J., Beule D., Malik A., Wolski E., Eickhoff H., Lehrach H. and Herzel H. Normalization strategies for cDNA microarrays. *Nucleic Acids Research*, Vol. 28(10), 2000.

[56] Shamir R. and Sharan R. Click: A clustering algorithm for gene expression analysis. In *In Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB '00). AAAI Press.*, 2000.

[57] Sherlock G. Analysis of large-scale gene expression data. *Curr Opin Immunol*, 12(2):201–205, 2000.

[58] Siedow, J. N. Meeting report: Making sense of microarrays. *Genome Biology*, 2(2):reports 4003.1–4003.2, 2001.

[59] Smet, Frank De, Mathys, Janick, Marchal, Kathleen, Thijs, Gert, Moor, Bart De and Moreau, Yves. Adaptive quality-based clustering of gene expression profiles. *Bioinformatics*, 18:735–746, 2002.

[60] Sokal, R.R. *Clustering and classification: Background and current directions. In Classifincation and clustering Edited by J. Van Ryzin*. Academic Press, 1977.

[61] Spellman PT. *et al.* Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol Biol Cell*, 9(12):3273–3297, 1998.

[62] Tamayo P., Solni D., Mesirov J., Zhu Q., Kitareewan S., Dmitrovsky E., Lander E.S. and Golub T.R. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci. USA*, Vol. 96(6):2907–2912, March 1999.

[63] Tang, C., Zhang, A., and Pei, J. Mining phenotypes and informative genes from gene expression data. In *In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'03)*, Washington, DC, USA, August 24-27 2003.

[64] Tang C., Zhang L., Zhang A. and Ramanathan M. Interrelated two-way clustering: An unsupervised approach for gene expression data analysis. In *Proceeding of BIBE2001: 2nd IEEE International Symposium on Bioinformatics and Bioengineering*, pages 41–48, Bethesda, Maryland, November 4-5 2001.

[65] Tang, Chun and Zhang, Aidong. An iterative strategy for pattern discovery in high-dimensional data sets. In *Proceeding of 11th International Conference on Information and Knowledge Management (CIKM 02)*, McLean, VA, November 4-9 2002.

[66] Tavazoie, S., Hughes, D., Campbell, M.J., Cho, R.J. and Church, G.M. Systematic determination of genetic network architecture. *Nature Genet*, pages 281–285, 1999.

[67] Tefferi, A., Bolander, E., Ansell, M., Wieben, D. and Spelsberg C. Primer on Medical Genomics Part III: Microarray Experiments and Data Analysis. *Mayo Clin Proc.*, 77:927–940, 2002.

[68] Thomas J.G., Olson J.M., Tapscott S.J. and Zhao L.P. An Efficient and Robust Statistical Modeling Approach to Discover Differentially Expressed Genes Using Genomic Expression Profiles. *Genome Research*, 11(7):1227–1236, 2001.

[69] Troyanskaya, O.,Cantor M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D. and Altman R. Missing value estimation methods for dna microarrays. *Bioinformatics*, In press.

[70] Tusher, V.G., Tibshirani, R. and Chu, G. Significance Analysis of Microarrays Applied to the Ionizing Radiation Response. *Proc. Natl. Acad. Sci. USA*, Vol. 98(9):5116–5121, April 2001.

[71] Wang, Haixun, Wang, Wei, Yang, Jiong and Yu, Philip S. Clustering by Pattern Similarity in Large Data Sets. In *SIGMOD 2002, Proceedings ACM SIGMOD International Conference on Management of Data*, pages 394–405, 2002.

[72] Wen, Xiling, Fuhrman, Stefanie, Michaels, George S., Carr, Daniel B., Smith, Susan, Barker, Jeffery L. and Smomgyi, Roland. Large-scale temporal gene expression mapping of central nervous system development. *Proc. Natl. Acad. Sci. USA*, Vol. 95:334–339, January 1998.

[73] Xing, E.P. and Karp, R.M. Cliff: Clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinformatics*, Vol. 17(1):306–315, 2001.

[74] Yang, Jiong, Wang, Wei, Wang, Haixun and Yu, Philip S. δ-cluster: Capturing Subspace Correlation in a Large Data Set. In *Proceedings of 18th International Conference on Data Engineering (ICDE 2002)*, pages 517–528, 2002.

[75] Yeung, Ka Yee and Ruzzo, Walter L. An empirical study on principal component analysis for clustering gene expression data. Technical Report UW-CSE-2000-11-03, Department of Computer Science & Engineering, University of Washington, 2000.

[76] Yeung, K.Y., Fraley, C, Murua, A., Raftery, AE., Ruzz WL. Model-based clustering and data transformations for gene expression data. *Bioinformatics*, 17:977–987, 2001.

[77] Yeung, K.Y., Haynor, D.R. and Ruzzo, W.L. Validating Clustering for Gene Expression Data. *Bioinformatics*, Vol.17(4):309–318, 2001.