

Cluster Analysis for Gene Expression Data: A Survey

Daxin Jiang and Aidong Zhang
Department of Computer Science and Engineering
State University of New York at Buffalo
Buffalo, NY 14260
{djiang3, azhang}@cse.buffalo.edu

Abstract

DNA microarray technology has made it now possible to monitor simultaneously the expression levels for thousands of genes during important biological processes (for example, cellular replication and the response to changes in the environment), and across collections of related samples (such as tumor samples from patients and normal persons). Elucidating the patterns hidden in the gene expression data is a tremendous opportunity for functional genomics. However, because of the large number of genes and the complexity of biological networks, it is also a huge challenge to comprehend and interpret the resulting mass of data, which often consists of millions of measurements. A first step in answering this challenge is via clustering techniques, which are essential in data mining process for exploring natural structure and identifying interesting distributions and patterns in underlying data.

In cluster analysis, one wishes to partition a given data set into groups based on given features such that the data points in a group are more similar to each other than points in different groups. There is a very rich literature on cluster analysis going back over three decades. Numerous approaches were proposed based on different quality criteria. Some of them have been applied to gene expression data, and have proved useful for identifying biologically relevant groupings of

genes and samples, and further helping answering such questions as gene function, gene regulation and gene expression differentiation under various conditions.

In this paper, we first give a brief introduction to DNA microarray technology and clustering techniques. We will then give a detailed description for a variety of clustering algorithms. We will also discuss several key issues in cluster analysis for gene expression data.

Index terms: microarray technology, gene expression data, cluster analysis.

1. Introduction

1.1 Introduction to Microarray Technology

1.1.1 Measuring mRNA levels

Compared with traditional approach to genomic research, which has been locally examining and collecting data on single gene, microarray technologies have made it now possible to monitor the expression pattern for tens of thousands of genes in parallel [Chen1998, Derisi1996, Ermolaeva1998, Heller1997, Iyer1999, Schena1995, Schena1996, Shalon1996, Welford1998]. A microarray is a glass slide, onto which tens of thousands of DNA molecules are attached at fixed locations (spots) where each spot relates to a DNA sequence. The two mRNA samples (or targets) are reverse-transcribed into cDNA, labeled using different fluorescent dyes (e.g. a red-fluorescent dye Cy5 and a green-fluorescent dye Cy3), then mixed and hybridized with the arrayed DNA sequences (or probes). After this competitive hybridization, the relative abundance of those spotted DNA sequences in two mRNA samples may be assessed by monitoring the

differential hybridization of the two samples to the sequences on the array. Afterwards, the slides are imaged using a scanner and fluorescence measurements are made separately for each dye at each spot on the array. There are also other methods to measure the mRNA levels such as Oligonucleotide chips, RT-PCR (Reverse Transcriptase Polymerase Chain Reaction), and SAGE (Serial Analysis of Gene Expression) [D'haeseleer1999]. Data collected by these methods are called *gene expression data*, which, after pre-processing, form the data source of our clustering algorithms.

1.1.2 From raw data to gene expression matrix

The raw data produced by microarrays are in fact monochrome images. Software provided with the image scanner identifies the spots corresponding to genes on the microarray, determines their boundaries, measures and compares the fluorescence intensity from each spot to the background intensity and to these intensities for other channels and produces two main quantities for each spot on the array: R and G, which are measures of the fluorescence intensities (transcript abundance) for the red and green labeled mRNA samples, respectively [Brazma2000]. These original gene expression data still come along with noise, missing values and systematic variations sourced from experiment procedure. Troyanskaya et al. [Troyanskaya2001] implemented and evaluated three methods for the estimation of missing values in gene expression data: a Singular Value Decomposition based methods (SVDimpute), weighted K-nearest neighbors (KNNimpute), and row average. Kerr et al. [Kerr 2001] stressed the importance of replication of the experiments in order to model the noise. At the same time, normalization, which is the process of identifying and removing systematic sources of variation and allowing between-slide comparisons has itself become an interesting research topic in the

filed of gene expression data analysis. Various approaches have been proposed to model the systematic variations before any further data analysis such as clustering can be applied [Kerr2000, Yang2001, Schuchhardt2000, Hill2001, Tseng2001].

After above pre-processing steps, gene expression data can be represented by a real-valued expression matrix X , where the rows of the matrix are vectors forming the expression patterns of genes, the columns of the matrix represent samples from either various conditions, development steps or different tissues and each cell x_{ij} is the measured expression level of gene i in sample j .

1.1.3 Applications of clustering gene expression data

Monitoring tens of thousands of genes in parallel under different experiment environment or across different tissue types provides a systematic genome-wide approach to solve the problems such as gene functions in various cellular process, gene regulations in various cellular signaling pathways and gene expression differentiation in various diseases or drug treatments [Brazma2000, D'haeseleer2000, Sharan2001]. In answer to all those questions, clustering technique manifests its crucial power as the first step in extracting information from the mass of gene expression data set.

Clustering algorithms have proved useful to help group together genes with similar functions based on gene expression patterns under various conditions or across different tissue samples [Alon1999, Tavazoie1999, Eisen1998, Shamir2000]. Co-expressed genes found in the same cluster demonstrate significant enrichment for function. By expanding functional families of genes with known function together with poorly characterized or novel genes may help understand the functions of many genes for which information is not previously available. It is

indicated a relatively small and redundant collection of conditions suffice to separate genes into functional categories, the addition of more and diverse conditions can only enhance those observation [Eisen1998]. Eisen et. al. applied a hierarchical clustering algorithm to two sets of data: gene expression of primary human fibroblasts stimulated with serum following serum starvation and gene expression in the budding yeast *Saccharomyces Cerevisiae* during time courses of mitotic cell division cycle, sporulation, the diauxic shift and shock response. By comparing the clustering results with the functional annotation in the *Saccharomyces* Genome Database, they found a strong tendency for those genes in the same cluster to share common roles in cellular processes and genes of unrelated sequence but similar function cluster tightly together. Similar tendency was also observed in human data [Eisen1998].

Gene clustering also becomes the first step to uncover the regulatory elements in transcriptional regulatory networks [Alon1999, Tavazoie1999, Xing2001]. Co-expressed genes in the same cluster are probably involved in the same cellular process and strong expression pattern correlation between those genes indicates co-regulation. By searching for common DNA sequences at the promoter regions of genes within the same cluster, regulatory motifs specific to each gene cluster are identified and cis-regulatory elements are proposed [Brazma1998, Roth1998, Helden1998, Wolfsberg1999, Tavazoie1999]. Tavazoie et. al. applied K-means algorithm to gene expression of *Saccharomyces cerevisiae* during two cell cycles. They grouped the genes into 30 clusters and searched for common upstream DNA sequence motifs within each cluster to identify cis-regulatory elements that may contribute to the co-regulation of genes in a cluster. They found 18 motifs from 12 different clusters, 7 of them had been identified experimentally and had been known to regulate the expression of many genes in their respective

clusters. Inference of regulation through clustering of gene expression data also warrants the further hypothesis of the mechanism of transcriptional regulatory network [Tavazoie1999].

In addition to gene expression data being analyzed by clustering genes using samples as features, clustering samples via genes as features also makes good sense as well. Clustering different samples based on gene expression is one of key issues in such problems as class discovery, normal and tumor tissue classification and drug treatment evaluation [Golub1999, Tavazoie1999, Xing2001]. Golub et. al. applied SOM clustering algorithm on gene expression data containing 6817 human genes of 38 leukemia samples (27 acute myeloid leukemia (ALL) and 11 acute lymphoblastic leukemia (AML)). SOM automatically groups the 38 samples into two classes with one containing 24 ALL out of 25 samples and the other 10 ALL out of 13 samples. They further used SOM to divide the samples into four classes. Subclasses of ALL, namely, B-lineage ALL and T-lineage ALL were distinguished. The author declared that class discovery techniques based on cluster analysis for gene expression data can be used to identify fundamental subtypes of any cancer [Golub1999].

1.2 Introduction to Clustering Techniques

1.2.1 Cluster and clustering

Clustering is the process of grouping the data into classes (or clusters) so that objects within a cluster have high similarity in comparison to one another, but are very dissimilar to objects in other clusters [Han2000]. Clustering is an example of unsupervised classification. By *classification*, we mean a procedure of assigning a data object to a set of classes; by

unsupervised, we mean clustering does not rely on predefined classes and class labeled training examples while labeling the data objects. Thus, clustering distinguishes itself from pattern recognition or the areas of statistics know as discriminant analysis and decision analysis, which seek to find rules for classifying objects given a set of pre-classified objects.

The definition of “cluster” is not precisely defined. In many different applications, the best definition depends on the type of data and the desired results. Several working definitions of clusters are commonly used [Barbara2000]:

- Well-separated clustering definition: A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.
- Center-based cluster definition: A cluster is a set of objects such that an object in a cluster is closer to the “center” (centroid) of a cluster, than to the center of any other cluster.
- Contiguous cluster definition: A cluster is a set of points such that a point in a cluster is closer to one ore more other points in the cluster than to any point not in the cluster.
- Density-based cluster definition: A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
- Similarity-based cluster definition: A cluster is a set of objects that are “similar” and objects in other clusters are not “similar”.

1.2.2 Procedure of a clustering task

The basic steps to develop clustering process can be summarized as follows:

- ◆ Feature selection

Feature selection is the process of identifying the most effective subset of the original features to use in clustering. Sometimes, one or more transformations may be applied to the original features to produce new salient features.

◆ Clustering process

This step refers to the application of a clustering algorithm to generate a good clustering scheme that fits the data set. Generally, a clustering algorithm defines a proximity measure and a search method to find the optimal or sub-optimal partition in the data object space according to some clustering criterion.

- Proximity measure is a measure that quantifies the similarity (or dissimilarity) of two data points.
 - Clustering criterion is the expression of our goal of clustering which is generally based on different working definition of a cluster and/or an expected distribution of underlying data in specific application domain.
- ◆ Cluster validation: Cluster validation is the assessment of a clustering scheme. Typically, validation indices are defined to assess the quality of clusters or to estimate the degree to which a clustering scheme fits a specific data set [Halkidi2000].

In the sequel, we will discuss the clustering issues of proximity measure and optimization criterion. We will discuss the problems of feature selection and cluster validation in the last section.

1.2.2.1 Proximity measure

In a proximity matrix P , the entry P_{ij} represents the proximity of the expression patterns for genes i and j (or expression profile for samples i and j). A good choice of measure depends on the nature of the biological question and on the technology that was used to obtain the data.

Euclidean distance is one of the most commonly used proximity measure for cluster analysis on gene expression data. The dissimilarity between the i th and j th objects is defined as:

$$D_{ij} = \sqrt{\sum_{l=1}^d (x_{il} - x_{jl})^2}$$

However, for gene expression data, patterns' similarity seems more important than their spatial distance. Pearson correlation coefficient is also widely used to measure the strength of the linear relationship between two objects. This measure has the advantage of calculating similarity depending only on the pattern but not on the absolute magnitude of the spatial vector [Tang2001]. The similarity between the i th and j th objects is defined as:

$$R_{ij} = \frac{\sum_{l=1}^d (x_{il} - \bar{x}_i)(x_{jl} - \bar{x}_j)}{\sqrt{\sum_{l=1}^d (x_{il} - \bar{x}_i)^2} \sqrt{\sum_{l=1}^d (x_{jl} - \bar{x}_j)^2}}$$

Using D_{ij} or R_{ij} as a measure of proximity has the problem that both are sensitive to outliers. An outlier could make D_{ij} higher than any bound or make R_{ij} equal to any value between -1 and 1 [Hubert1981]. Spearman's rank-order correlation coefficient, which replaces numeric gene expression level x_{ij} by its rank within corresponding row or column, is more robust to outliers [Bickel2001]. Other measures used include Euclidean distance between expression profiles and

slopes, squared Pearson correlation, Euclidean distance between pair-wise correlations to all other genes and mutual information [D'haeseleer2001].

1.2.2.2 Optimization criterion

Many clustering algorithms are performed by minimizing or maximizing some criterion (objective function) based on the chosen measure of proximity. For example, partitioning-based algorithms such as K-means seek to minimize the sum of the distance of an object from the “center” of the cluster. Hierarchical algorithms do not have global objective functions, they make decisions at each step based on the local objective functions such as single link, complete link, group average, wards method, centroid method [Barbara2000]. Each of them defines the proximity between two clusters in a different way. Model-based algorithms assume that the data is a “mixture” of a number of underlying statistical distributions. The criterion for those algorithms is to maximize the likelihood of estimate for the statistical parameters that describe the clusters. Different optimization criteria reflect different understanding of cluster and different assumptions of underlying data model. Optimization criterion, together with proximity measure, constitutes the fundamental elements from which a clustering algorithm is derived.

1.2.3 Criteria for evaluating clustering algorithms

Considering the characters of gene expression data, and the particular applications in functional genomics, the optimal algorithms for analysis of gene expression data need the following properties [Jain1999, Barbara2000, Han2000]:

- Scalability and efficiency: Algorithms should be efficient and scalable considering the large amount of data to be handled.

- Irregular shape: Algorithms need to be able to identify a dense set of points which forms a cloud of irregular nonspherical shapes, including those with lacunae or concave sections and nested shapes, as a cluster.
- Robustness: The clustering mechanisms should be robust against large amounts of noise and outlier.
- Order insensitivity: Algorithms should not be sensitive to the order of input. That is, clustering results should be independent of data order.
- Cluster number: The number of clusters inside the data set needs to be determined by the algorithm itself and not prescribed by the user.
- Parameter estimation: The algorithms should have the ability to estimate any parameters required by the algorithm from the data set, and no *a priori* knowledge of the data or domain knowledge input is required from the user.
- Dimensionality: Algorithms need the ability to handle data with high dimensionality or the ability to find clusters in subspaces of the original space.
- Stability: No data object will be classified into different clusters for different running of the algorithm.
- Incrementability: Algorithms should be able to incrementally handle the addition of new data or the deletion of old data instead of re-running the algorithms on the new data set.
- Interpretability: The clustering results of the algorithms need to be interpretable. That is, clustering may need to be tied up with specific biological interpretations and applications.

2. Clustering Algorithms

2.1 K-means

The K-means algorithm is a typical partitioning-based clustering method. Given a set of N objects, a partitioning method constructs K partitions of the data, where each partition represents a cluster and $K \leq N$ [Han2000]. The K-means algorithm is derived by asking how we can obtain a partition of the data which optimizes the following objective function:

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

In the above equation, p is a data object, m_i is the center (centroid) of cluster C_i , and K is an input parameter as the number of clusters. Thus, the criterion function E attempts to minimize the sum of the squared distance of an object from the “center” of the cluster.

The K-means algorithm begins by K initial selected data objects, either randomly or manually by user, as cluster *centroids*. Next, each data object is assigned to the cluster with the nearest centroid. Then the centroid for each cluster is recalculated as the mean of all data objects belonging to the cluster. This process iterates until no more changes occur, or the amount of change falls below a pre-defined threshold.

The K-means method is simple and commonly used in analyzing DNA microarray data. In the absence of numerical problems, this procedure always converges very fast (typically 5-10 iterations) to a solution. However, K-means needs as input parameters the cluster number K and initial centroids. Randomly chosen K and initial centroids can lead to poor results. Also, K-means objective function is minimized by globular clusters of equal size or by clusters that are

well separated [Barbara2000]. Hence, K-means algorithm only works well for finding spherical-shaped clusters in small to medium-sized data set. In order to achieve global optimality in partitioning-based clustering, it would require the exhaustive enumeration of all possible partitions. K-means algorithm adopts a heuristic which represents each cluster by the mean value of the objects in the cluster. Thus it does not guarantee a globally optimized solution.

2.2 SOM

The Self-Organizing Map (SOM) was developed by Kohonen [Kohonen1984] based on a single layered neural network. The data points are present at the input and the output neurons are organized with a simple neighborhood structure, e.g. a two dimensional $k \times l$ grid. Each neuron 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 point 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.

The SOM clustering algorithm begins with the initialization of the reference vectors. The training process first randomly selects a training sample (data point), then determines the neuron with the closest reference vector to the current point and updates this reference vector and the reference vectors for all neighboring neurons. This process iterates until the reference vectors are not changing much. When the training is complete, the algorithm assigns all points to the closest neurons and returns the clusters.

Suppose the reference vector of neuron N at iteration i is denoted as $f_i(N)$. The initial mapping f_0 is random. On subsequent update steps, a data point P is selected and the neuron N_p to which P is “closest” mapped is identified. The mapping of neuron is then adjusted by moving towards P via the following equation:

$$f_{i+1}(N) = f_i(N) + \tau(d(N, N_p), i)(P - f_i(N)).$$

The learning rate τ decreases with distance of neuron N from N_p and with iteration number i . Typically, τ is chosen as either Gaussian function or step function [Barbara2000].

SOM gives an intuitively appealing map of the high-dimensional data set and proves helpful in visualizing high-dimensional gene data into a 2-D or 3-D space. However, the learning update procedures of SOM are quite similar to those in some classical clustering approaches, such as K-means (The fundamental conceptual difference is that during the training process, SOM uses each data point to update not only the closest reference vector but also the reference vectors of nearby neurons). SOM does not overcome the problems of K-means such as cluster number determination, sub-optimization, and tendency to detect only spherical clusters. Furthermore, its convergence is controlled by various parameters such as the learning rate and the grid topology of the neurons.

2.3 Hierarchical Clustering

In contrast to partitioning-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, namely, *dendrogram*. The branches of a dendrogram not only record the formation of the clusters but also indicates 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* algorithms and *divisive* algorithms 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 of the groups are merged into one cluster. Divisive algorithms (top-down approach) start 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 measurements of cluster proximity derive various merge strategies known as single link, complete link, group average, wards method, and centroid method [Barbara2000]. For divisive approaches, we need to decide which clusters to split at each step. Some are based on heuristic methods such as deterministic annealing algorithm [Alon1999], while many others are based on graph theoretic methods which we will discuss later.

Similar to heuristic methods, hierarchical clustering suffers from the fact that the data objects are grouped based on local decisions, with no guarantee of global optimization. This problem is even exacerbated by the deterministic nature of hierarchical clustering that once a step is done, it can never be undone. Hierarchical clustering has also been noted by statisticians to have trouble with one or more of the following: lack of robustness, non-uniqueness, non-convex shape, and a tendency to break large clusters. Furthermore, strict phylogenetic trees are best suited to situations

of true hierarchical descent (such as in the evolution of species) but are not designed to reflect the multiple distinct ways in which expression patterns can be similar [Tomayo1999].

2.4 Graph Theoretic Approach

Given a dataset X with *proximity matrix* P , we can construct a weighted graph $G(V,E)$, called *proximity graph*, where each data point corresponds to a vertex. For some methods, each pair of objects is connected by an edge with assigned weight which equals to or is derived from the proximity value between the objects [Shamir2000, Xing2001], while for others, the proximity can be mapped to either 0 or 1 by some threshold, and edges only exist between objects i and j where $P[i j]$ equals 1 [Ben-Dor1999, Hartuv1999]. Although the hierarchical clustering algorithms we discussed in the previous section are most commonly viewed in terms of merging or splitting clusters, they can also be viewed as operations on a proximity graph. Graph theoretic clustering techniques are explicitly cast in terms and concepts of a graph, and converts the problem of clustering a dataset into some graph theoretic problems such as finding minimum cut or maximal cliques in the proximity graph G .

2.4.1 HCS, CLICK and CLIFF

Hartuv et al. [Hartuv1999] presents the naïve HCS algorithm which seeks to identify *highly connected components* in proximity graph as cluster. The basic HCS algorithm takes as input the proximity graph G . First, it computes a minimum cut of G , which has the minimal number of edges whose removal results in a disconnected graph. If the cut meets some criterion (e.g. the edge number of the cut is greater than half of vertex number), G is returned and claimed as a cluster. Otherwise, it is split into its two most loosely connected pieces according to this

minimum cut. Afterwards, each piece goes through the same process recursively. The algorithm stops when each data point either belongs to some cluster or be a singleton.

CLuster Identificatoin via Connectivity Kernals (CLICK) [Shamir2000] is based on the same idea of HCS algorithm. However, it makes the probabilistic assumption that after normalization, pair-wise similarity values between elements (no matter they are in the same cluster or not) are normally distributed. Under this assumption, the weight w_{ij} of an edge (i,j) is defined to reflect the probability that i and j are in the same cluster. CLICK also expands the HCS algorithm by an *adoption step* which handles the remaining singletons and updates the current clusters and a *merging step* which iteratively merges two clusters with similarity exceeding a predefined threshold. Several ad-hoc refinements are also developed to increase the scalability of the algorithm.

In [Xing2000], the authors indicates that CLICK has little guarantee not going astray and generating partitions highly unbalanced. Thus sophisticated pruning techniques need to be developed to explicitly enforce cut balance. Clustering via Iterative Feature Filtering (CLIFF) [Xing2000] addresses the unbalance problem based on the concept of normalized cut. For any two (not necessarily disjoint) subsets A and B of the vertex set V , they define the weight

$$w(A, B) = \sum_{u \in A} \sum_{v \in B} w(u, v).$$

In the normalized cut framework, the bias toward highly unbalanced cuts is eliminated by normalizing $w(A, \bar{A})$ relative to $w(A, V)$ and to $w(\bar{A}, V)$. Specifically, the *normalized weight* of the cut A, \bar{A} is defined as follows:

$$Ncut(A, \bar{A}) = \frac{w(A, \bar{A})}{w(A, V)} + \frac{w(\bar{A}, \bar{A})}{w(\bar{A}, V)} .$$

An *optimal normalized cut* is a cut of minimum normalized weight.

2.4.2 CAST

Ben-Dor et al. [Ben-Dor1999] introduced the idea of a *corrupted clique graph* data model. The input data is assumed to come from the underlying cluster structure by “contamination” with random errors caused by the complex process of gene expression measurement. To be specific, it is assumed that the true clustering of the data points can be represented by a *clique graph* H , which is a disjoint union of complete graphs, with each clique corresponding to a cluster. The similarity graph G is derived from H by flipping each edge/non-edge with probability α . Therefore, to clustering a dataset equals to identify the original clique graph H given the corrupted version G with as few flips (errors) as possible.

In [Ben-Dor1999], the authors presented both a theoretic algorithm, for which they proved some performance results, and a practical heuristic, call CAST (Cluster Affinity Search Technique). CAST is based on the same ideas as the theoretical one, but it does not require the number of clusters and avoids enumeration of all possible cluster candidates.

CAST takes as input a real symmetric n -by- n similarity matrix S and an affinity threshold t , where $0 \leq t \leq 1$. The algorithm constructs the clusters one at a time. The currently constructed cluster is denoted by C_{open} . Each element x has an affinity value with respect to C_{open} defined as $a(x) = \sum_{y \in C_{open}} S(x, y)$. An element has *high affinity* if it satisfies $a(x) \geq t|C_{open}|$. Otherwise, x has

low affinity. 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 one new cluster at a time until all elements have been assigned to a cluster.

The obvious advantage of CAST algorithm is that the number of clusters is determined by the algorithm and no prior knowledge of the cluster structure is required. It is based on a stochastic model for cluster formation and for data errors, so that it can be analyzed probabilistically. Also, the algorithm employs an incremental strategy: it first identifies a high-quality subset of elements as the “core” of a cluster, then it generates and adjusts the complete cluster with less complex method. However, it is also important to note that the affinity threshold and error rate indirectly influence the cluster structure. Moreover, there is no normal proof of time complexity or even convergence of the heuristic.

2.5 Model-based Clustering

Clustering algorithms based on statistical *mixture models* offer a principled alternative to heuristic algorithms [Yeung2001a, Fraley2001]. Model-based clustering assumes that the data is generated by a finite mixture of underlying probability distributions with each component corresponding to a different cluster. The framework supposes the data set X consists of independent multivariate observations x_1, \dots, x_n and an underlying set of k unknown distributions E_1, E_2, \dots, E_k . Let $f_i(x_r | \theta)$ denote the density of an observation x_r with respect to E_i , where θ is the set of unknown parameters, and τ_r^i represents the probability that x_r belongs to E_i . Each data point is constrained to belong to some cluster. So $\tau_r^i \geq 0$ and $\sum_{i=1}^k \tau_r^i = 1$. Given these notations,

the goal of the scheme is to find the parameters θ and τ that maximize the likelihood for the mixture model:

$$L_C(\theta_1, \dots, \theta_k; \tau_1, \dots, \tau_n | x) = \prod_{r=1}^n \sum_{i=1}^k \tau_r^i f_i(x_r | \theta_i).$$

Generally, each component E_i is modeled by the multivariate normal distribution with parameters μ_i (mean vector) and Σ_i (covariance matrix).

EM (*expectation-maximization*) algorithm is a standard process to estimate the values of μ_i , Σ_i , and τ_r . The EM algorithm iterates between an *E-step* where hidden parameters ($\hat{\tau}_r$) are conditionally estimated from the data with the current parameter estimates ($\hat{\mu}_i, \hat{\Sigma}_i$) and an *M-step* where the model parameters ($\hat{\mu}_i, \hat{\Sigma}_i$) are estimated to maximize the complete-data likelihood given the hidden parameters.

One of the advantages of the mixture-model scheme is that it provides a systematic mechanism to determine the number of clusters and compares the parameterization of different models. An approximate to *Bayes factor*, called *Bayesian Information Criterion (BIC)*, is applied to address the problem of model selection. To avoid the criticism for weakness to noise and outliers, the framework also models noise and outliers as a constant-rate Poisson process with intensity ν which distributes points throughout the input space.

In practice, the basic model-based procedure starts with noise detection and elimination. Next, some agglomerative hierarchical clustering is applied to obtain an initial partition of the data set,

based on which the EM algorithm is used to refine the parameters for the *Gaussian components*. Finally, the *BIC* is used to select the model that best fits the data.

Model-based clustering relies on the assumption that the data fits a *Gaussian distribution*. This may not be true in many cases. In particular, modeling gene expression data sets is an ongoing effort by many researchers. To our knowledge, there is no well-established model to represent gene expression data yet. Although preprocessing, such as Box-Cox transformation and standardized transformation [Yeung2001a], are applied to raw gene expression data, it is still questionable to which extent gene expression data satisfy the Gaussian mixture assumption. Furthermore, the convergence rate of EM algorithm can be very slow and it may not be practical for models with very large numbers of components [Fraley2001], and no attempt is made to analyze the time or space required for the algorithm as a function of input size [Fasulo1999].

2.6 PCA/SVD

Principle component analysis (PCA), often performed by *singular value decomposition (SVD)*, is a classical, statistical method to capture the dominant system variation of the data set. Let X be a gene expression data set with n genes and p experimental conditions. To cluster the genes in the data set, the experimental conditions (columns) are the variations. Let \vec{x}_j be a column vector of the expression levels of all the n genes under experimental condition j . A *PC* is a linear transformation of the experimental conditions. Let $\vec{z}_k = \sum_{j=1}^p \alpha_{k,j} \vec{x}_j$ be the k th *PC*, Σ be the covariance matrix of the data, and $\vec{\alpha}_k$ be a column vector of all the $\alpha_{k,j}$'s, i.e. $\vec{\alpha}_k^T = (\alpha_{k,1}, \alpha_{k,2}, \dots, \alpha_{k,p})$. The k th *PC* \vec{z}_k can be derived by maximizing $\text{var}(\sum_{j=1}^p \alpha_{k,j} \vec{x}_j)$, such that

$\vec{\alpha}_i^T \vec{\alpha}_k = 1$ and $\vec{\alpha}_i^T \vec{\alpha}_k = 0$, where $i < k$. It can be shown that $\vec{\alpha}_k$ is the eigenvector corresponding to the k th largest eigenvalue λ_k , and $\text{var}(\vec{z}_k) = \lambda_k$ [Jolliffe1986]. From the derivation of PC 's, the k th PC can be interpreted as the direction that maximizes the variation of the projections of the data points such that it is orthogonal to the first $k-1$ PC 's, and the k th PC has the k th largest variance among all PC 's.

Hastie et al. [Hastie2000] introduced a statistical method based on PCA called *gene shaving* with the aim to extract groups of elements (genes) that have coherent expression and vary as much as possible across the samples. The basic idea is to project the genes from sample space into PC space with one dimension at a time and with a sequence from high to low.

The *gene shaving* algorithm starts with preprocessing the entire expression matrix X by centering each row to have zero mean. Next, it computes the leading principal component of the rows of X and the proportion α (typically 10%) of the genes that have smallest absolute inner-product with the leading principal component considered non-significant and discarded ("shaved off"). This process repeats until only one gene remains and produces a nested sequence of gene clusters $S_N \supset S_k \supset S_{k1} \supset S_{k2} \dots \supset S_1$, where S_k denotes a cluster of k genes. Then a statistical test by *Gap Statistic* is used to estimate and select the optimal cluster size \hat{k} . The algorithm continues to find the next cluster after orthogonalizing each row of X with respect to $\bar{x}_{S_k}^{\wedge}$, the average gene in S_k^{\wedge} . This process continues until a maximum of M clusters are found, where M is chosen *a priori*.

By applying *PCA* to gene expression data, some underlying patterns that characterize the features of genes may be uncovered. Genes with strong correlation to some pattern are grouped together and regarded as participating in the same cellular process. However, genes may participate several cellular processes at the same time, and the overall expression value of a gene may be contributed by those multiple cellular processes. That means genes could be in multiple clusters, or clusters are not necessarily to be exclusive with each other. Projecting the overall expression value into all *PC*'s found in the data set may better help identify all those processes that gene plays a role in. *Gene shaving* only makes use of the first *PC* and generates clusters one by one, thus clusters are exclusive because genes inside previous identified clusters have already been “shaved off”.

2.7 Two-way Clustering

Information in gene expression matrices is special in that it can be studied in two dimensions [Brazma2000]: analyzing expression patterns of genes by comparing rows in the expression profiles matrix and analyzing expression profiles of samples by comparing columns in the matrix. All clustering algorithms mentioned above are one-way clustering techniques, i.e. clustering either genes or samples. Although some of them can be applied to both genes and samples, they are typically performed on these two axes separately, i.e. one after another. In the following, we will introduce several clustering algorithms that clustering on both gene-dimension and sample-dimension simultaneously. We call these algorithms *two-way clustering*.

The two-way clustering methods above are based on the belief that only a small subset of the genes participates in any cellular process of interest. Moreover, it is also believed that any cellular process takes place only in a subset of the samples, which suggests a cluster of genes be

defined with respect to only a subset of samples. Furthermore, a single gene may participate in multiple pathways that may or may not be coactive under all conditions. This indicates a gene to be in more than one cluster, or in none at all, i.e., partitions may not be exclusive and exhaustive. These observations induced a series of approaches based on *block clustering* [Hartigan1972], where “block” in those approaches means a subset of genes with coherent expression of patterns and large variation across a subset of samples.

2.7.1 Basic block clustering

Tibshirani et al. [Tibshirani1999] added a backward pruning procedure on the block splitting algorithm by Hartigan [Hartigan1972] and devised a permutation-based method for deciding on the optimal number of blocks called *Gap Statistics*. Hierarchical representation of clusters can be achieved by restricting the splits to only those that intersect the blocks. The block clustering begins with the entire data in one block. For each step, it chooses the row or column which splits some existing block into two pieces and produces largest reduction in the total within block variance. This process continues until a large number of blocks are obtained. Finally, it recombines the split blocks until the optimal number of blocks are obtained.

2.7.2 Biclustering

Cheng et al. [Cheng2000] introduced a similarity score called *mean squared residue* to measure the coherence of the genes and conditions in the block, called *bicluster*. A low mean squared residue score plus a large variation from the constant form a good criterion for identifying a block.

Given a subset I of genes and a subset J of samples, the mean squared residue score of a submatrix is defined as follow:

$$H(I, J) = \frac{1}{|I| |J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2$$

where

$$a_{iJ} = \frac{1}{|J|} \sum_{j \in J} a_{ij}, \quad a_{Ij} = \frac{1}{|I|} \sum_{i \in I} a_{ij}$$

and

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

are the row and column means and the mean in the submatrix A_{IJ} . A submatrix A_{IJ} is called a δ -bicluster if $H(I, J) \leq \delta$ for some $\delta \geq 0$. The lowest score $H(I, J) = 0$ indicates that the gene expression levels fluctuate in unison. This includes the trivial or constant clusters where there is no fluctuation. The row variance may be an accompanying score to reject trivial bicluser:

$$V(I, J) = \frac{1}{|J|} \sum_{j \in J} (a_{ij} - a_{Ij})^2.$$

The brute force method to find a bicluster is by computing the score H for each possible row/column addition/deletion and choose the action that decreases H the most. If no action decreases H , or if $H \leq \delta$, a bicluster is returned. Cheng et al. applied a greedy approach to reduce the complexity down to $O(mn)$, where n and m are the row and column sizes of the expression matrix.

2.7.3 Plaid model

Lazzaroni et al. [Lazzaroni2000] proposed a *plaid model* to fit the gene expression data. The idea is to regard the gene expression data as a sum of multiple terms called 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. Thus, each layer is actually a “block”.

The plaid model is formalized as the following:

$$Y_{ij} = \mu_0 + \sum_{k=1}^K (\mu_k + \alpha_{ik} + \beta_{jk}) \rho_{ik} \kappa_{jk},$$

where μ_0 is the gene expression level of background, μ_k describes expression level in cluster k , α_{ik} represents expression level of a set of genes that had an identical, though not constant responding to a set of samples, and β_{jk} represents expression levels of a set of samples with a common expression pattern for a set of genes. ρ_{ik} is 1 if gene i is in the k th gene-block (zero otherwise), and κ_{jk} is 1 if sample j is in the k th sample-block (zero otherwise). Since overlap is allowed, $\sum_k \rho_{ik} \geq 2$ for some i , or $\sum_k \kappa_{jk} \geq 2$ for some j is allowed.

To best fit the model to the data, the following criterion Q is minimized through an iterate approach:

$$Q = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^p (Z_{ij} - (\mu + \alpha_i + \beta_j) \rho_{ik} \kappa_{jk})^2.$$

They adopt an iterative approach with each cycle updating μ , α , β values, ρ_{ik} values and κ_{jk} values in turn. The iteration process is quite similar to EM algorithm discussed in Model-based

clustering to estimate the parameters and improve the objective function along one direction at a time till at last a local minimum is reached.

2.7.4 CTWC

Coupled two-way clustering (CTWC) [Getz2000] seeks to identify couples of relatively small subset of features (F_i) and objects (O_j), where both F_i and O_j can be either genes or samples, such that when using only the features in F_i to cluster corresponding objects O_j , stable and significant partitions emerge. CTWC provides a heuristic to avoid brute-force enumeration of all possible combinations: only subset of genes or samples that are identified as stable clusters in previous iteration are candidates for the next iteration.

CTWC begins with only one pair of genes and samples (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 *super-paramagnetic clustering algorithm (SPC)*, is applied on each set and 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 (G for genes and S for samples), and a list of pairs of gene and sample subsets (g_n^i, s_m^j). For each iteration, one gene subset from G and one sample subset from S that have not been combined before are coupled and clustered mutually as objects and features. Newly generated stable clusters are added into G and S , and a pointer that identify the parent pair is recorded to indicate where this cluster comes from. The iteration continues until no new clusters that satisfy some criteria, such as stability and critical size, are found.

3. Discussion

3.1 Determine the Number of Clusters

One of the key issues in clustering analysis is to determine the number of clusters in the data set. Intervention by the users with specific domain knowledge can greatly help enhance the clustering performance. Some algorithms, like *K-means*, *SOM*, and *gene shaving*, require pre-assigned number of clusters as one of the input parameters. In [Tavazoie1999], 30 was chosen as the number of clusters to partly reflect human's pre-concept of the data set. However, those algorithms are obviously not suitable for some exploratory tasks when little or no previous knowledge of the data set is available.

Ideally, the number of clusters is identified by the algorithm itself based on the model it imposes on the data set and the criterion it specifies for a cluster. For example, graph theoretic approaches look upon clusters in a data set as *highly connected components* or *maximal cliques* in a proximity graph. When all such structures in the graph are identified, all clusters and spontaneously, the number of clusters are also uncovered. However, the number of clusters is still indirectly influenced by input parameter, e.g. *affinity threshold*, which has to be assigned by the user. Moreover, in such complex data set as gene expression data, clusters can be of any arbitrary shapes and sizes and well separated clusters hardly exist. Model hypothesis, and corresponding criterion definition assumed by the clustering algorithms are not necessarily compatible with the true structure of underlying data set. Consequently, the algorithm performance could be very poor.

Some statistical approaches such as *Bayesian Model Selection* and *Gap statistic* defined some statistic with respect to the number of clusters. By providing systematic methods to estimate and compare a series of parameters of the data model, they are more flexible compared with the graph theoretic algorithms which return fixed number of clusters. Fraley et al. proposed an approximate *Bayes factors* based on their mixture model framework to compare models with different parameters. The statistic, called *Bayesian Information Criterion (BIC)* is determined by two terms: first the post probability that the data conform to the model M and the second one is the number of independent parameters to be estimated in the model. Since the more parameters in the model, the better the mixture model fits a given data set, the second term is used to penalize the complexity of the model. *BIC* is defined as the following:

$$BIC \equiv 2\mathbf{lm}(\mathbf{x}, \hat{\theta}) - m_M \log(n),$$

where n is the number of data objects, $\mathbf{lm}(\mathbf{x}, \hat{\theta})$ is the maximized mixture likelihood for the model M and m_M is the number of independent parameters to be estimated in the model. Accordingly, the model with the largest value of *BIC* would be selected as the optimal one.

If we plot the *error message* W_k (within cluster dispersion) versus the number of clusters k , we will find W_k decreases monotonically as the number of clusters k increases, but the decrease flattens markedly from some k . Such k is considered a good indication for the “true” number of clusters in the data set. The *Gap Statistic* [Tibashirani2000] provides a statistical procedure to detect this “elbow” by comparing the log value of W_k to its expectation under an appropriate null reference distribution of the data. The procedure begins with clustering the observed data with various pre-assigned number of clusters, $k=1,2,\dots,K$, and computing the within dispersion

measures W_k for each k . Then it generates B reference datasets based on some reference distribution, clustering each reference datasets and computes the within dispersion measure W_{kb}^* , $b=1,2,\dots,B$, $k=1,2,\dots,K$. Define *Gap Statistic* as:

$$Gap(k) = (1/B) \sum_b \log(W_{kb}^*) - \log(W_k).$$

Compute the standard deviation

$$sd_k = [(1/B) \sum_b (\log(W_{kb}^*) - \bar{l})^2]^{1/2},$$

where $\bar{l} = (1/B) \sum_b \log(W_{kb}^*)$,

and define

$$s_k = sd_k \sqrt{1 + 1/B}.$$

Finally, the number of clusters is determined via

$$\hat{k} = \text{smallest } k \text{ such that } Gap(k) \geq Gap(k+1) - s_{k+1}.$$

From above procedure, we can see k is the first point where w_k falls behind its expectation value far enough. So k is the estimated ‘‘elbow’’.

To determine the number of clusters is a crucial but still open problem in cluster analysis. For a comprehensive survey of methods for estimating the number of clusters, see [Milligan1985], while [Gordon1999] discusses the best performances.

3.2 Reduce the Gene Dimension

Gene expression data typically have the following characters:

- High dimensionality of the gene space, which is currently $10^3 \sim 10^4$ and would be $10^5 \sim 10^6$ in near future with the development of microarray technology.
- Great difference of dimensionality between the sample space and the gene space ($10^1 \sim 10^2$ samples versus $10^3 \sim 10^4$ genes)
- Large amount of irrelevant or redundant features in gene space and high correlation in both gene space and sample space.

When clustering samples using genes as features, we have to face tens of thousands of features, and those inherent characters of gene expression data cause the following difficulties:

- It is well known as *the curse of dimensionality* that the more features one needs to model, the harder the modeling task becomes, because the size of the search space increases exponentially with the number of parameters of the model.
- With the existence of irrelevant and redundant features, a clustering algorithm is hard to partition the samples corresponding to actual empirical interest, because the discriminating pattern of interested genes may be either overruled by other patterns that represent the gender, age, or other disease variability, or inundated by the distance calculation from the large number of irrelevant features.

Various methods have been taken to select features for microarray sample clustering. Statistical methods are used for identifying differentially expressed genes in both within-slide and multiple-slide experiment results to filter out genes that do not change much during the experiment [Baggerly2001, Baldi2001, Butte2001, Chen1997, Claverie1999, Dudoit2000, Efron2000, Herwig2001, Ideker2000, Manducchi2000, Newton2001, Park2001, Theilhaber2001, Thomas

2001, Tsien2001, Wittes1999, Xiong2001, Yue2001]. Another approach first groups the features into coherent sets using a clustering algorithm and then projects the samples onto a lower-dimensional space spanned by the average expression patterns of the coherent feature sets [Hastie2000]. Raychaudhuri et al. [Raychaudhuri2000] applied principle component analysis to gene expression data (where the experimental conditions are the variables and the gene expression measurements are the observations) to define a core set of independent features for the expression states that allow them to be compared directly.

Several approaches have been proposed to dynamically use the relationship between the groups of samples and genes and combine a clustering process and a feature selection process iteratively. Suppose we are given a set of predefined classes and a classification of samples, i.e., a reference partition. We can score each gene according to its relevance to the reference partition such as *t-test score* and *separation score* and select a set of genes with highest relevance as qualified features. Unfortunately, in cluster analysis, such previous knowledge and training data are not available. However, when we inspect the relationship between the two problems, sample clustering and gene selection, we can find that they are actually two sides of one coin. If we can find informative genes, then it is relatively easy to use traditional clustering algorithms to clustering samples. On the other hand, if we can correctly partition the samples, informative genes can be found by sorting all genes according to their relevance to the partition. The intuition is that although we do not know the exact target partition *a priori*, with respect to which we would like to optimize the feature subset, at each iteration we can expect to obtain an approximate partition that is close to the target one, and thus allows the selection of an

approximately good feature subset, which will hopefully draw the partition even closer to the target partition in the next iteration.

Xing et al. [Xing2001] proposed CLIFF (clustering of high-dimensional microarray data via iterative feature filtering). The algorithm first uses the unsupervised independent feature modeling technique to rank all features in terms of their discriminability. Then it generates an initial partition based on the k most discriminative features, where k is specified in advance. Based on this partition, feature selection is treated roughly as a “supervised” learning problem, where information gain ranking and Markov blanket filter can be applied, and the newly determined features subset can be used to generate a new partition, which can be used to further improve the feature selection.

Tang et al. [Tang2001] proposed an interrelated two-way clustering approach for unsupervised analysis of gene expression data. The approach is to delineate the relationships between gene clusters and sample partitions while conducting an iterative search for sample patterns and detecting significant genes of empirical interest. During the iteration process, they use the relationships of sample clusters and gene groups thus discovered to post a partial or approximate pattern. Then this pattern is used to direct the elimination of irrelevant genes. In turn, the remaining meaningful genes will guide further sample pattern detection. The criterion for terminating the series of iterations is determined by evaluating the quality of the sample partition.

3.3 Evaluate the Clustering Results

We have introduced several different clustering algorithms, all of which partition the data set from different points of views and tend to emphasize different types of regularities in the data. As a consequence, different clustering algorithms or even the same algorithm along with different input parameters may yield very different partitions given the same data set. Moreover, since there is no precise definition of ‘what a cluster is’, the “best” partition of the data set typically depends on the particular application at hand. Thus, to evaluate the results of a clustering algorithm is as important as the algorithm itself for a biologist in face of a particular application and a specific gene data set.

If we have a reference partition P of the data set, which is probably derived from previously known domain knowledge, we can simply evaluate the cluster result C by comparing the similarity between P and C through some statistic such as *Folks and Mallars index*, *Huberts F statistic* [Halkidi2000], or computing for each cluster the probability of observing at least k objects form one of the predefined partitions such as *P -value* [Tavazoie1999]. However, this kind of validation is basically used to compare the performance of different algorithms because such reference partition typically does not exist in advance.

Other approaches measure the quality of generated clusters from the concept of “homogeneity and separation”. Several validity indices, such as *Dunn and Dunn-like indices* [Dunn1974, Pal1997, Theodoridis1999], *Davis Bouldin index* [Davies1979, Pal1997], and *SD index* [Halkidi2000b], are also defined to measure to what degree the data objects are similar inside one cluster, while dissimilar between different clusters.

A novel approach to evaluate the cluster quality is based on the following logic: If putative classes reflect true structure, then a class predictor based on those classes should perform well [Golub1999]. To assess the generated clusters, one or part of the data objects are left out as test samples. Only the remaining are used for clustering. Several statistics such as *figure of merit*, *prediction strength* are defined to indicate the predictive power of the clustering results. Typically, the process of validation uses cross-validation, i.e., data objects are used in turn as test samples and the capacity of cluster algorithm to meaningfully group together similar data objects is quantized by the distribution of those statistic [Golub1999, Yeung2001c, Tibshirani2001].

Yeung et al [Yeung2001c] introduced *figure of merit (FOM)* for assessing the quality of clustering results. They applied a clustering algorithm to all but one sample e , and estimate the predictive power by measuring the within-cluster similarity of e . Their intuition is that the tendency of similar expression level for genes in the same cluster indicates a biological significant clustering. Thus the greater the within-cluster similarity of e , the stronger the predicative power and thereby the better the clustering scheme. Let $R(g,e)$ be the expression level of gene g under condition e in the raw data matrix. Let $\mu_{c_i}(e)$ be the average expression level in condition e of genes in cluster c_i . The 2-norm figure of merit, $FOM(e,k)$, for k clusters using condition e as validation 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 used as the left-out sample. The *aggregate figure of merit*, $FOM(k) = \sum_{e=1}^m FOM(e,k)$, is an estimate of the total predictive power of the algorithm over all

the samples for k clusters in a data set. To compensate the influence that simply increasing the number of clusters will tend to decrease the FOM , they also define the adjusted figure of merit.

Tibshirani et al. [Tibshirani2001] divided all the samples into two parts, namely, training sample set X_{tr} and test sample set X_{te} . Obviously, X_{tr} and X_{te} follow the same data distribution. The main idea is to use the clustering results (e.g. cluster center) generated from the training sample to predict the “co-membership” (whether two samples are in the same cluster) in the test sample set. The procedure is as follows:

- i) Cluster the test data X_{te} into k clusters $A_{k1}, A_{k2}, \dots, A_{kk}$. Denote the clustering operation by $C(X_{te}, k)$ and “co-memberships” by $D[C(X_{te}, k), X_{te}]_{ii'} = 1$ if observations i and i' fall into the same cluster, and zero otherwise.
- ii) Cluster the training data X_{tr} into k clusters
- iii) Measure how well the training set cluster centers predict co-memberships in the test set.

They define the *prediction strength* of the clustering $C(\cdot, k)$ by

$$ps(k) = \min_{1 \leq j \leq k} \frac{1}{n_{kj}(n_{kj} - 1)} \sum_{i \neq i' \in A_{kj}} I(D[C(X_{tr}, k), X_{te}]_{ii'} = 1)$$

For each test cluster, they compute the proportion of observed pairs in that cluster that are also assigned to the same cluster by cluster centers derived from training samples. The prediction strength is the minimum of this quantity over the k test clusters. It can be seen that the higher value of ps , the better quality the clustering scheme has.

Notice that the *aggregate FOM* and ps defined above are both statistic with respect to k , the number of clusters. Thus, when applied to clustering results generated by the same algorithm

with different pre-assigned cluster number, they can also help determine the optimal number of clusters in the data.

3.4 Visualize High Dimensional Data

Visualization of the data set is an important part of cluster analysis and is a crucial verification of the clustering results. However, effective visualization would be difficult in the case of large multi-dimensional gene expression data set. Hierarchical clustering algorithms graphically represent the data by *dendrogram* and the data set can be organized in certain order so that the branches of the corresponding *dendrogram* do not cross and thus arranges the ‘similar’ data object placed near each other [Eisen1999]. *SOM* maps the high dimensional data into 3- or 2-D space with certain assumption of topology among the input objects. *PCA* captures the maximal variations in the data set with a set of ordered variables (*PC's*), which are linear transformations of the original set of variables. By projecting the data points on the first few *PC's*, *PCA* reduces the dimensionality of the data set while retaining as much as possible the variation in the data set. Zhang et al. presented a dynamic interactive visualization environment, VizCluster, and its application on clustering gene expression data [Zhang2002]. VizCluster combines the merits of both high dimensional projection scatter-plot and parallel coordinate plots. The primary idea is a nonlinear projection which maps the n-dimensional vectors into two-dimensional points. To preserve the information at different scales and yet reduce the typical problem of parallel coordinate plots being messy caused by overlapping lines, a zip zooming viewing method is proposed [Zhang2002]. Visualization tools for gene expression data provide simple, fast, intuitive and yet powerful views of the data set. The applications include the classification of samples, evaluation of gene clusters and cluster detection and validation.

References

- [Alon1999] U. Alon, N. Barkai, D.A. Notterman, K.Gish, S. Ybarra, D. Mack, and A.J. Levine. "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide array", Proc. Natl. Acad. Sci. USA, Vol. 96(12):6745-6750, June 1999.
- [Alter2000] O. Alter, P.O. Brown, and D. Bostein. "Singular value decomposition for genome-wide expression data processing and modeling", Proc. Natl. Acad. Sci. USA, Vol. 97(18):10101-10106, August 2000.
- [Baldi2001] P. Baldi and A.D. Long. "A Bayesian framework for the analysis of microarray expression data: regularized t -test and statistical inferences of gene changes", Bioinformatics 2001 17: 509-519.
- [Baggerly2001] K.A. Baggerly, K.R. Coombes, K.R. Hess, D.N. Stivers, L.V. Abruzzo and W Zhang. "Identifying Differentially Expressed Genes in cDNA Microarray Experiments", Journal of Computational Biology 8(6), 639-659, 2001.
- [Barbara2000] D. Barbara. "An Introduction to Cluster Analysis for Data Mining", <http://www.ise.gmu.edu/~dbarbara/755/csurvey.pdf>.
- [Beissbarth2000] T. Beissbarth, K. Fellenberg, B. Brors, et al. "Processing and quality control of DNA array hybridization data", Bioinformatics 16, 1014-1022, 2000.
- [Ben-Dor1999] A. Ben-Dor, R. Shamir and Z. Yakhini. "Clustering Gene Expression Patterns", Proceedings of the Third Annual International Conference on Computational Molecular Biology (RECOMB'99), Lyon, France, Aug. 11-14, 1999.
- [Bickel2001] D.R. Bickel. "Robust Cluster Analysis of DNA Microarray Data: An Application of Nonparametric Correlation Dissimilarity", Medical College of Georgia, July 2001.
- [Blatt1996] M. Blatt, S.Wiseman, and E. Domany. Phys. Rev. Lett. 76, 3251-3255, 1996.
- [Brazma1998] A. Brazma, I. Jonassen, J. Vilo and E. Ukkonen. "Predicting Gene Regulatory Elements in Silico on a Genomic Scale", Genome Research, 8, 1202-1215, 1998.
- [Brazma2000] A. Brazma and J. Vilo. "Minireview: Gene expression data analysis", Federation of European Biochemical societies, 480:17-24, June 2000.
- [Butte2001] A.J. Butte, J. Ye, G. Niederfellner, K. Rett, H.U. Häring, M.F.White, and I. S. Kohane. "Determining Significant Fold Differences in Gene Expression Analysis", Pacific Symposium on Biocomputing 6:6-17, 2001.
- [Cheesman1996] P. Cheesman and J. Stutz. "Bayesian classification (AutoClass): theory and results", In Advances in Knowledge Discovery and Data Mining, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (eds.), Cambridge, MA: AAAI/MIT press, pp. 153-180 1996.

- [Chen1997] Y. Chen, E.R. Dougherty and M.L. Bittner. "Ratio based decisions and the quantitative analysis of cDNA microarray images", *Journal of Biomedical Optics* 2(4), 364-374, 1997.
- [Chen1998] J.J. Chen, R. Wu, P.C. Yang, J.Y. Huang, Y.P. Sher, M.H. Han, W.C. Kao, P.J. Lee, T.F. Chiu, F. Chang, Y.W. Chu, C.W. Wu and K. Peck. "Profiling expression patterns and isolating differentially expressed genes by cDNA microarray system with colorimetry detection", *Genomics*, Vol.51:313-324, 1998.
- [Cheng2000] Y. Cheng and G.M. Church. "Biclustering of expression data", ISMB'00, 2000.
- [Chudin2001] E. Chudin, R. Walker, A. Kosaka, S. Wu, D. Rabert, T.K. Chang and D.E. Kreder. "Assessment of the relationship between signal intensities and transcript concentration for Affymetrix GeneChip arrays", *Genome Biology*, 3(1): research0005.1-0005.10, 2001.
- [Chu1998] S. Chu, J. Derisi, M. Eisen, J. Mulholland, D. Bostein, P.O. Brown and I. Herskowitz. "The transcriptional program of sporulation in budding yeast", *Science* 282, 699-705, 1998.
- [Claverie1999] J.M. Claverie. "Computational methods for the identification of differential and coordinated gene expression", *Human Molecular Genetics*, Vol. 8, No. 10, 1821-1832, 1999.
- [Davies1979] D.L. Davies and D.W. Bouldin. "A cluster separation measure", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 1(2), 1979.
- [Derisi1996] J. DeRisi, L. Penland, P.O. Brown, M.L. Bittner, P.S. Meltzer, M. Ray, Y. Chen, Y.A. Su and J.M. Trent. "Use of a cDNA microarray to analyse gene expression patterns in human cancer", *Nature Genetics*, Vol.14:457-460, 1996.
- [Diebolt1994] J. Diebolt and C.P. Robert. "Bayesian estimation of finite mixture distributions", *J. R. Stat. Soc. B*, 56, 363-375, 1994.
- [D'haeseleer1999] P. D'haeseleer, S. Liang, and R. Somogyi. "Tutorial: Gene Expression Data Analysis and Modeling", *Pacific Symposium on Biocomputing*, January 1999.
- [D'haeseleer2000] P. D'haeseleer, S. Liang, and R. Somogyi. "Genetic network inference: from co-expression clustering to reverse engineering", *Bioinformatics*, Vol.16(8):707-726, 2000.
- [Dudoit2000] S. Dudoit, Y.H. Yang, M.J. Callow, and T.P. Speed. "Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments", *Technical report 578*, Stanford University, Department of Biochemistry Stanford University School of Medicine, August 2000.
- [Dunn1974] J.C. Dunn. "Well separated clusters and optimal fuzzy partitions", *J. Cybern.* Vol. 4, pp 95-104, 1974.

- [Efron2000] B. Efron, R. Tibshirani, V. Goss and G. Chu. "Microarrays and Their Use in a Comparative Experiment", Tech. report, Stanford University, 2000.
- [Eisen1998] M.B. Eisen, P.T. Spellman, P.O. Brown and D. Botstein. "Cluster analysis and display of genome-wide expression patterns", Proc. Natl. Acad. Sci. USA, Vol. 95:14863-14868, December 1998.
- [Ermolaeva1998] O. Ermolaeva, M. Rastogi, K.D. Pruitt, G.D. Schuler, M.L. Bittner, Y. Chen, R. Simon, P. Meltzer, J.M. Trent and M.S. Boguski, "Data management and analysis for gene expression arrays", Nature Genetics, Vol.20:19-23, 1998.
- [Fasulo1999] D. Fasulo. "An Analysis of Recent Work on Clustering Algorithm", April 26, 1999.
- [Fraley1998] C. Fraley and A.E. Raftery. "How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis", Technical Report No. 329. Department of Statistics University of Washington, 1998.
- [Getz2000] G. Getz, E. Levine and E. Domany. "Coupled two-way clustering analysis of gene microarray data", Proc. Natl. Acad. Sci. USA, Vol. 97(22):12079-12084, October 2000.
- [Golub1999] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gassenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, D.D. Bloomfield, and E.S. Lander. "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring", Science, Vol. 286(15):531-537, October 1999.
- [Gordon1999] A. Gordon. Classification (2nd edition), Chapman and Hall/CRC press, London, 1999.
- [Halkidi2000] M. Halkidi, Y. Batistakis and M. Vazirgiannis. "Clustering algorithms and validity measures", 2000.
- [Halkidi2001a] M. Halkidi, Y. Batistakis and M. Vazirgiannis. "On Clustering Validation Techniques", 2001.
- [Halkidi2001b] M. Halkidi, M. Vazirgiannis and I. Batistakis. "Quality scheme assessment in the clustering process", 2001.
- [Halkidi2001c] M. Halkidi and M. Vazirgiannis. "Clustering Validity Assessment: Finding the optimal partitioning of a data set", 2001.
- [Han2000] J. Han and M. Kamber. "Data Mining: Concepts and Techniques", The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor Morgan Kaufmann Publishers, August 2000. ISBN 1-55860-489-8.

- [Hartigan1972] J. Hartigan. "Direct clustering of a data matrix", J. Amer. Statis. Assoc. 6, 123-129, 1972.
- [Hartuv1999] E. Hartuv, A. Schmitt, J. Lange, S. Meier-Ewert, H. Lehrach and R. Shamir. "An Algorithm for Clustering cDNAs for Gene Expression Analysis", Third International Conference on Computational Molecular Biology (RECOMB) 1999.
- [Hastie2000] T. Hastie, R. Tibshirani, M.B. Eisen, A. Alizadeh, R. Levy, L. Staudt, W.C. Chan, D. Botstein and Patrick Brown. "Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns", Genome Biology, Vol. 2(1):0003.1-0003.21, August 2000.
- [Helden1998] J.V. Helden, B. Andre and J. Collado-Vides. "Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies", J Mol Biol, 281, 827-842, 1998.
- [Heller1997] R.A. Heller, M. Schena, A. Chai, D. Shalon, T. Bedilion, J. Gilmore, D.E. Woolley and R.W. Davis. "Discovery and analysis of inflammatory disease-related genes using cDNA microarrays", Proceedings of the National Academy of Sciences of the United States of America, Vol.94:2150-2155, 1997.
- [Herwig2001] R. Herwig, P. Aanstad, M. Clark and H. Lehrach. "Statistical evaluation of differential expression on cDNA nylon arrays with replicated experiments", Nucleic Acids Research, Vol. 29, No. 23 e117, 2001.
- [Hill2001] A.A. Hill, E.L. Brown, M.Z. Whitley, G. Tucker-Kellogg, C.P. Hunter and D.K. Slonim. "Evaluation of normalization procedures for oligonucleotide array data based on spiked cRNA controls", Genome Biology, 3(1): research0055.1-0055.13, 2001.
- [Holter2000] N.S. Holter, M. Mitra, A. Maritan, M. Cieplak, J.R. Banavar and N.V. Fedoroff. "Fundamental patterns underlying gene expression profiles: Simplicity from complexity", Proc. Natl. Acad. Sci. USA, Vol. 97(15):8409-8414, July 2000.
- [Hubert1981] P.J. Hubert. Robust Statistics, John Wiley & Sons (New York), 1981.
- [Ideker2000] T. Ideker, V. Thorsson, A.F. Siegel, L.E. Hood. "Testing for Differentially-Expressed Genes by Maximum-Likelihood Analysis of Microarray Data", Journal of Computational Biology 7(6), 805-818, 2000.
- [Iyer1999] V.R. Iyer, M.B. Eisen, D.T. Ross, G. Schuler, T. Moore, J.C.F. Lee, J.M. Trent, L.M. Staudt, Jr. J. Hudson, M.S. Boguski, D. Lashkari, D. Shalon, D. Botstein and P.O. Brown. "The transcriptional program in the response of human fibroblasts to serum", Science, Vol.283:83-87, 1999.
- [Jain1999] A.K. Jain, M.N. Murty, P.J. Flynn. "Data Clustering: A Review". ACM Computer Surveys, Vol. 31, No.3, 1999.

- [Jartiv1999] E. Jartiv, A. Schmitt, J. Lange, S. Meier-Ewert, H. Lehrach and R. Shamir. "An Algorithm for Clustering cDNAs for Gene Expression Analysis", REOMB 1999.
- [Jolliffe1986] I.T. Jolliffe. *Principal Component Analysis*. New York: Springer-Verlag, 1986.
- [Kerr2000] M.K. Kerr, M. Martin and G.A. Churchill. "Analysis of variance for gene expression microarray data", *Journal of Computational Biology*, 7:819-837, 2000.
- [Kerr2001] M.K. Kerr and G. Churchill. "Experimental design for gene expression microarrays", *Biostatistics*, 2:183-201, 2001.
- [Kohonen1984] T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, Berlin, 1984.
- [Kolatch2001] E. Kolatch. "Clustering Algorithms for Spatial Databases: A Survey".
- [Laan2000] J. Mark, V.D. Laan, and J.F. Bryan. "Gene Expression Analysis with the Parametric Bootstrap", University of California, Berkeley June 27, 2000.
- [Lazzeroni2000] L. Lazzeroni and A. Owen. "Plaid models for gene expression data", Dept Statistics, Stanford University, March 2000.
- [Manducchi2000] E. Manducchi et al. "Generation of patterns from gene expression data by assigning confidence to differentially expressed genes", *Bioinformatics*, Vol.16, no. 8, 685-698, 2000.
- [Milligan1985] G.W. Milligan and M.C. Cooper. "An Examination of Procedures for Determining the Number of Clusters in a Data Set", *Psychometrika*, 50, 159-179, 1985.
- [Newton2001] M.A. Newton, C.M. Kendzioriski, C.S. Richmond, F.R. Blattner and K.W. Tsui. "On Differential Variability of Expression Ratios: Improving Statistical Inference about Gene Expression Changes from Microarray Data", *Journal of Computational Biology* 8(1), 37-52, 2001.
- [Park2001] P.J. Park, M. Pagano and M. Bonetti. "A Nonparametric Scoring Algorithm for Identifying Informative Genes from Microarray Data", *Pacific Symposium on Biocomputing* 6:52-63, 2001.
- [Raychaudhuri2000] S. Raychaudhuri, J.M. Stuart, and R.B. Altman. "Principal components analysis to summarize microarray experiments: application to sporulation time series", In *Pacific Symposium on Biocomputing*, pages 415-426, 2000.
- [Roth1998] P. Roth, J.D. Hughes, P.W. Estep and G.M. Church. "Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation", *Nature Biotechnology*, 16, 939-945, 1998.

[Schena95] M. Schena, D. Shalon, R.W. Davis and P.O. Brown. "Quantitative monitoring of gene expression patterns with a complementary DNA microarray", *Science*, Vol.270:467-470, 1995.

[Schena96] M. Schena, D. Shalon, R. Heller, A. Chai, P.O. Brown and R.W. Davis. "Parallel human genome analysis: microarray-based expression monitoring of 1000 genes", *Proceedings of the National Academy of Sciences of the United States of America*, Vol.93:10614-10619, 1996.

[Schuchhardt2000] J. Schuchhardt, D. Beule et al. "Normalization strategies for cDNA microarrays", *Nucleic Acids Research*, Vol. 28, No.10, e47, 2000.

[Sclove1983] S.C. Sclove. "Application of the conditional population mixture model to image segmentatin". *IEEE Trans. Patt. Anal. Mach. Intell.*, PAMI-5, 428-433, 1983.

[Shamir2000] R. Shamir and R. Sharan. "Click: A clustering algorithm for gene expression analysis." In *proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB '00)*. AAAIPress, 2000.

[Shalon1996] D. Shalon, S.J. Smith and P.O. Brown. "A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization", *Genome Research*, Vol.6:639-645, 1996.

[Sharan2001] R. Sharan, R. Elkon and R. Shamir. "Clustering Analysis and its Application to Gene Expression Data", 2001.

[Sharma1996] S.C. Sharma. *Applied Multivariate Techniques*, John Willwy & Sons, 1996.

[Smyth1996] P. Smyth. "Clustering using Monte Carlo Cross-Validation". *KDD-96*, pp.126—133, 1996.

[Spellman1998] P.T. Spellman, G. Sherlock, M.Q. Zhang, V.R. Iyer, K. Anders, M.B. Eisen, P.O. Brown, D. Bostein and B. Futcher. "Comprehensive identification of cell cycle-related genes of the yeast *saccharomyces* by microarray hybridization", *Mol. Biol. Cell* 9, 3273-3297, 1998.

[Tamayo1999] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E.S. Lander and T.R. Golub. "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.

[Tang2001] C. Tang, L. Zhang and A. Zhang. "Interrelated Two-way Clustering: an Unsupervised Approach for Gene Expression Data Analysis", 2001.

- [Tavazoie1999] S. Tavazoie, J.D. Hughes, M.J. Campbell, R.J. Cho and G.M. Church. "Systematic determination of genetic network architecture", *Nature genetics*, volume 22 , pages 281-285, 1999.
- [Theilhaber2001] J. Theilhaber, S. Bushnell, A. Jackson and R. Fuchs. "Bayesian Estimation of Fold-Changes in the Analysis of Gene Expression: The PFOLD Algorithm", *Journal of Computational Biology* 8(6), 585-614, 2001.
- [Theodoridis1999] Y. Theodoridis. "Spatial Datasets: an "unofficial" collection", <http://dias.cti.gr/~ythead/research/datasets/spatial.html>, 1999.
- [Thomas2001] J.G. Thomas, J.M. Olson, S.J. Tapscott and L.P. Zhao. "An Efficient and Robust Statistical Modeling Approach to Discover Differentially Expressed Genes Using Genomic Expression Profiles", *Genome Research* 11:1227-1236, 2001.
- [Tibshirani1999] R. Tibshirani, T. Hastie, M. Eisen, D. Ross, D. Botstein and P. Brown. "Clustering methods for the analysis of DNA microarray data", Stanford University, October 1999.
- [Tibshirani2000] R. Tibshirani, G. Walther and T. Hastie. "Estimating the number of clusters in a dataset via the Gap statistic", 2000.
- [Tibshirani2001] R. Tibshirani, G. Walther, D. Bostein and P. Brown. "Clustering validation by prediction strength", 2001.
- [Titterington1985] D.M. Titterington, A.F.M. Smith, U.E. Makov. "Statistical Analysis of Finite Mixture Distributions", Chichester, UK: John Wiley and Sons, 1985.
- [Troyanskaya2001] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein and R.B. Altman. "Missing value estimation methods for DNA microarrays", *Bioinformatics* Vol. 17 no. 6 2001, Pages 520-525, 2001.
- [Tseng2001] G.C. Tseng, O. MK, L. Rohlin, J.C. Liao, W.H. Wong. "Issues in cDNA microarray analysis: quality filtering, channel normalization, models of variations and assessment of gene effects", *Nucleic Acids Res.* 2001 Jun 15;29(12):2549-57.
- [Tsien2001] C.L. Tsien, T.A. Libermann, X. Gu, and I.S. Kohane. "On Reporting Fold Differences", *Pacific Symposium on Biocomputing* 6:496-507, 2001.
- [Wall2001] M.E. Wall, P.A. Dych and T.S. Brettin. "SVDMAN—Singular value decomposition analysis of microarray data", 2001.
- [Welford1998] S.M. Welford, J. Gregg, E. Chen, D. Garrison, P.H. Sorensen, C.T. Denny and S.F. Nelson. "Detection of differentially expressed genes in primary tumor tissues using representational differences analysis coupled to microarray hybridization", *Nucleic Acids Research*, Vol.26: 3059-3065, 1998.

- [Wittes1999] J. Wittes, H.P. Friedman. "Searching for Evidence of Altered Gene Expression: a Comment on Statistical Analysis of Microarray Data", Journal of the National Cancer Institute, Vol. 91, No. 5, 400-401, March 3, 1999.
- [Wolfsberg1999] T.G. Wolfsberg., A.E. Gabrielian, M.J. Campbell, R.J. Cho, J.L. Spouge and D. Landsman. "Candidate regulatory sequence elements for cell cycle-dependent transcription in *saccharomyces cerevisiae*", Genome Research, 9, 775-792, 1999.
- [Xie1991] X.L. Xie and G. Beni. "A Validity measure for Fuzzy Clustering", IEEE Transactions on Pattern Analysis and machine Intelligence, Vol. 13, No4, August 1991.
- [Xing2001] E.P. Xing and R.M. Karp. "CLIFF: Clustering of High-Dimensional Microarray Data via Iterative Feature Filtering Using Normalized Cuts", Bioinformatics DISCOVERY NOTE, Vol.1(1):1-9, 2001.
- [Xiong2001] M. Xiong, X. Fang, J. Zhao. "Biomarker identification by feature wrappers", Genome Research, 11:1878-1887, 2001.
- [Yang2000] Y.H. Yang, S. Dudoit, P. Luu and T.P. Speed. "Normalization for cDNA Microarray Data", Tech.report, University of Berkeley, December 2000.
- [Yeung2001a] K.Y. Yeung, C. Fraley, A. Murua, A.E. Raftery and W.L. Ruzzo. "Model-based clustering and data transformations for gene expression data", University of Washington, April 2001.
- [Yeung2001b] K.Y. Yeung and W.L. Ruzzo. "An empirical study on principal component analysis for clustering gene expression data", Technical Report UW-CSE-01-04-02, University of Washington, 2001.
- [Yeung2001c] K.Y. Yeung, D.R. Haynor and W.L. Ruzzo. "Validating clustering for gene expression data". Bioinformatics, Vol.17(4):309-318, 2001.
- [Yue2001] H. Yue, P.S. Eastman. "An evaluation of the performance of cDNA microarrays for detecting changes in global mRNA expression", Nucleic Acids Research, Vol.29, No. 8 e41, 2001.
- [Zhang2002] L. Zhang, C. Tang, Y. Shi, Y. Song, A. Zhang, and M. Ramanathan, "VizCluster: An Interactive Visualization Approach to Cluster Analysis and Its Application on Microarray Data", proceedings of the Second SIAM International Conference on Data Mining (SDM'2002), Arlington, Virginia, April 11-13, pp. 19-40, 2002.