

Gene Expression Profile Classification: A Review

Musa H. Asyali^{*}, Dilek Colak¹, Omer Demirkaya² and Mehmet S. Inan³

¹Department of Biostatistics, Epidemiology, and Scientific Computing, King Faisal Specialist Hospital and Research Center, P.O. Box 3354, MBC-03, Riyadh, 11211 Saudi Arabia

²Department of Biomedical Physics, King Faisal Specialist Hospital and Research Center P.O. Box 3354, MBC-03, Riyadh, 11211 Saudi Arabia

³Department of Genetics, King Faisal Specialist Hospital and Research Center, P.O. Box 3354, MBC-03, Riyadh, 11211 Saudi Arabia

Abstract: In this review, we have discussed the class-prediction and discovery methods that are applied to gene expression data, along with the implications of the findings. We attempted to present a unified approach that considers both class-prediction and class-discovery. We devoted a substantial part of this review to an overview of pattern classification/recognition methods and discussed important issues such as preprocessing of gene expression data, curse of dimensionality, feature extraction/selection, and measuring or estimating classifier performance. We discussed and summarized important properties such as generalizability (sensitivity to overtraining), built-in feature selection, ability to report prediction strength, and transparency (ease of understanding of the operation) of different class-predictor design approaches to provide a quick and concise reference. We have also covered the topic of biclustering, which is an emerging clustering method that processes the entries of the gene expression data matrix in both gene and sample directions simultaneously, in detail.

1. INTRODUCTION

Due to recent advances in DNA microarray technology, it is now feasible to obtain gene expression profiles of tissue samples at relatively low costs. Many scientists around the world use the advantage of this gene profiling to characterize complex biological circumstances and diseases. Microarray techniques that are used in genome-wide gene expression and genome mutation analysis help scientists and physicians in understanding of the pathophysiological mechanisms, in diagnoses and prognoses, and choosing treatment plans.

Transcriptional profiling is a tool that provides unique data about disease mechanisms, regulatory pathways, and gene function [1]. This technology not only allows comparison of gene profiles in normal and pathological tissues or cells, but also helps us establish interrelationships among genes, e.g. clustering of genes, coincident temporal pattern of expression, identify upstream and downstream targets of genes, understand mechanisms of disease at a molecular level, and define and validate novel drug targets.

By using currently available commercial tools, a single experiment using this microarray technology can now provide systematic quantitative information on the expression of over 45,000 human transcripts within cells in any given state, enabling the investigator to inquire the whole genome at once. Although there are several different methods for massively parallel measurement of gene expression data, two methods have become widely accepted. The first, known as the *spotted cDNA microarray*

technology, is pioneered at the Stanford University. This technology involves robotic spotting of aliquots of purified cDNA clones, PCR products from clones or oligonucleotides onto glass slides that can contain thousands of arrayed elements [2]. Major advantages of this approach include the ability to design and build custom arrays that suit particular experiments or studies. In this technology, specific cDNA or oligonucleotide libraries from a species or a tissue of interest can be generated and arrayed. Alternatively, specific oligonucleotides, Expressed Sequence Tag (EST) and gene clones can be ordered directly or else amplified by Polymerase Chain Reaction (PCR). However, because of the mechanical nature of the arraying technology, the spotted microarray technology is quite operator-dependent, labor intensive, and the spotted microarray technology requires many months of fine-tuning for the system to work efficiently. Although signal strength may not be reproducible from one experiment to the next, due to variations in probe spotting and hybridization conditions, most scanners can detect two different emission wavelengths allowing normalization to a reference target.

The second approach, developed by the Affymetrix, Inc. (www.affymetrix.com), employs photo-lithography, the technology used in the manufacturing of computer chips, for embedding DNA probes on silicon chips. Affymetrix GeneChip™ microarrays consist of 25 base-pair long probes (25-mers). Each probe has a complementary probe with a central base mismatch to provide a measure of non-specific hybridization/binding and therefore serves as one of several internal controls. Genes are represented by typically 10-20 probes. Therefore, Affymetrix GeneChip™ is different from spotted cDNA microarrays in that all the Affymetrix GeneChips from the same family are identical. This helps reduce the variance due to changing probe amounts. However, using multi probes to measure the expression

*Address correspondence to this author at the Department of Computer Engineering, Faculty of Engineering and Architecture, Yasar University, Sehıtler Caddesi, 1522 Sokak, No: 6, Alsancak, Izmir, Turkey; Tel: + 90 232 463 33 44; Fax: +90 232 463 07 80; E-mail: asyali@ieec.org

levels of genes, introduces other problems. First and foremost, it is observed that the probes for a particular gene do not behave similarly (e.g., as the target amount increases, the response of different probes do not increase with the same rate). Therefore, combining the information picked by probes behaving differently is a major challenge and this issue remains to be a topic of intensive research. Li and Wong [3] and Irizarry *et al.* [4] have introduced/suggested different modeling approaches to cope with this problem. The underlying idea in these seemingly different approaches is to use many arrays/chips at the same time for each biological condition, instead of Affymetrix's own way of analyzing GeneChip data, which is based on single chips. Incorporating many arrays at the same time allows for the modeling of probe behavior. Literature shows that, despite its problems, the GeneChip system can be applied to both gene expression analysis and genomic mutation analysis.

A comprehensive review of biological and technological aspects of microarray technology can be found in [5]. Ramaswamy *et al.* [6] and Alizadeh *et al.* provide [7] a detailed discussion of the clinical implications of microarrays in oncology. For excellent reviews on many different aspects of microarray technology, the reader is referred to the two special supplements [8, 9]. References [10-14] provide an overview of gene expression data analysis. Topics covered include experimental design issues, normalization, quality control, exploratory analysis (data visualization), and the problem of multiple testing for determining the differentially expressed genes. Aittokallio *et al.* [15] and Quackenbush [16] underlined that the methods used to analyze the gene expression data can have a profound influence on the interpretation of the results and therefore a basic understanding of bioinformatics tools is required for optimal experimental design and meaningful data analysis.

Availability of gene expression profiles of tissue samples from different diagnostic classes led to the application of many well-established pattern recognition/classification algorithms to these profiles, in an attempt to provide more accurate and automatic diagnostic class prediction [7, 14, 17-28]. Compared to conventional tissue classification techniques like visual inspection under light-microscope, advantages of this type of classification, based on statistical analysis, are many-fold. Firstly, these are quantitative and systematic approaches; secondly, recent studies report relatively high class-prediction accuracies using these methodologies, showing that the idea of diagnostic class prediction based on gene expression profiles is realistic. Furthermore, some approaches due to their built-in gene selection capability, can pinpoint relatively important or influential genes in class prediction, which in turn may lead to the discovery of genes that are responsible for certain conditions. Some parametric (i.e., model based) approaches can also indicate the certainty of class predictions in terms of probabilities, therefore one can rely on the automatic predictor's response if the prediction certainty is very high, and may choose to look at or further investigate the cases for which class prediction strength is low. This feature of predictors based on gene expression profiles may lead to new diagnostic screening tests with higher cost-efficiency, throughput, and prediction accuracy.

There has been a dramatic increase in the number of papers/studies that involve gene expression profile classification. Fig. 1 shows the number articles that we found been in the PubMed Central¹ (<http://www.ncbi.nlm.nih.gov/entrez/query.fcgi>) by using the keywords "gene expression" AND {"classification" OR "clustering"} by year (from 1998 up to 2004). One can appreciate the exponential growth in the interest in this particular field of research. A comprehensive review discussing even the most important points that have been highlighted in these thousands of studies is a daunting task. Furthermore, statistical pattern recognition or classification is a relatively well-established field in terms of research. In contrast, the analysis of gene expression data, whether it is for the discovery of important genes or for the development diagnostic or prognostic prediction tools, has many issues that need to be resolved. For instance, standardization is an important issue. Here, we will briefly touch upon this issue.

Brazma *et al.* [29] and Ball *et al.* [30] discussed the importance of establishing a standard for recording and reporting microarray-based gene expression data and proposed a Minimum Information About a Microarray Experiment (MIAME) that describes the minimum information required to ensure that microarray data can be easily interpreted and that results derived from its analysis can be independently verified. Kuo *et al.* [31] compared two high-throughput cDNA microarray technologies, Stanford type (i.e., spotted) cDNA microarrays and Affymetrix oligonucleotide microarrays and showed that corresponding mRNA measurements from the two platforms showed poor correlation. Further, their results suggest gene-specific, or more precisely, probe-specific factors influencing measurements differently in the two platforms, implying a poor prognosis for a broad utilization of gene expression measurements across platforms. In another study, Nimgaonkar *et al.* [32] studied the reproducibility of gene expression levels across two generations of Affymetrix GeneChips and concluded that although experimental replicates are highly reproducible, the reproducibility across generations depends on the degree of similarity of the probe sets and the expression level of the corresponding transcript. If we consider the fact that even gene probes are not standardized and/or well defined yet, we can better understand the seriousness of the issues surrounding the microarray technologies.

Therefore, writing a survey on the issue of gene expression profile classification at this time is further complicated by these challenges. However, we believe that an attempt to have the snapshot of the current status of this remarkable field of study would still be beneficial for the greater community of scientists who embark on the study of gene expression data.

This review discusses various classification approaches applied to microarray gene expression data and implications of the corresponding results. It does not discuss the issue of biological experiment design using microarrays in this review. Although, this issue is extremely important, it is beyond the focus/interest. The reader is referred to excellent

¹PubMed® is the U.S. National Library of Medicine's (NLM™) web-based journal literature search system.

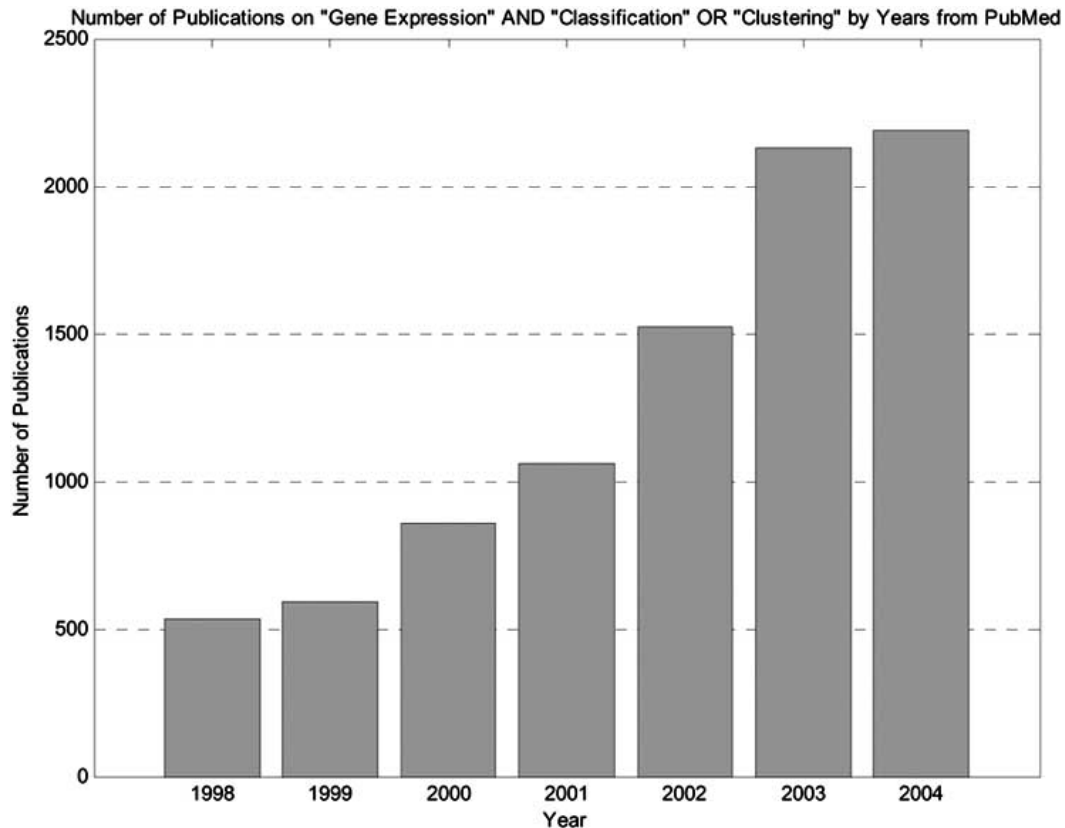


Fig. (1). The number of articles reported by PubMed search engine by using the keywords “gene expression,” AND {“classification” OR “clustering”} by year (from 1998 up to 2004).

reviews on this particular topic [33-37], where important considerations and potential pitfalls for microarray experiments are clearly highlighted. The important issue of sample size determination is discussed in [38]. Therefore, it is assumed that the biological/microarray study that gave rise to gene expression profiles is designed in such a manner that the underlying biological variance exceeds the technological variance and the variance due to other sources of error. If the overall noise is so high that it overshadows the biological variance, a classification study on such noisy data would not be reliable at all. For both spotted microarrays and Affymetrix arrays, the typical sources of unwanted variance include variations in RNA purity or quantity, washing efficiency, and scanning efficiency. In addition, the spotted microarrays suffer from the variance due to different labeling efficiencies of fluorescent dyes and uneven spotting of cDNA onto array surface.

The organization of the paper is as follows. Section 2 will present a general background/overview of pattern classification/recognition and touch upon important issues such as curse of dimensionality, gene/feature selection, measuring or estimating classifier performance. While covering these issues classification (predictor-design) will be reviewed studies on gene expression profiles in literature in some detail. Section 3 will be devoted to clustering studies on gene expression data, as the literature has many successful applications of this type of analysis. Following Jain *et al.* [39], *clustering* will be considered/treated under the general umbrella of classification methods. Clustering,

like all other classification methods, can *learn* or extract information from the training samples. The major difference that separates clustering from other classification methods is that it is *unsupervised*. Clustering aims at finding natural groupings in the data, as such, the class labels of training samples need not to be known for it to operate. (Some clustering approaches do not even need to be supplied with the number of classes/groups within the data.) While the underlying methodology for clustering and other classification methods (i.e., supervised learning) have a lot in common, there is a clear distinction between the uses of supervised learning and clustering approaches. Therefore, special emphasis will be given to clustering methods in a separate section and their use on microarray gene expression data will be discussed. This distinction is highlighted in Fig. 2, where we provide a global look at gene expression data analysis for class-prediction and class-discovery purposes.

Section 4 discuss *biclustering* as an emerging clustering method that processes/shuffles the entries of the gene expression data matrix in both directions simultaneously to find the natural clusters in gene and sample directions. Finally Section 5, will present concluding remarks.

Following the most commonly used format in the literature, it is assumed that the gene expression data used in the classification/clustering study is arranged in a p by n tabular (matrix) format where rows and columns correspond to genes and samples, respectively. (Typically, p and n are on the order of 10,000 and 100.)

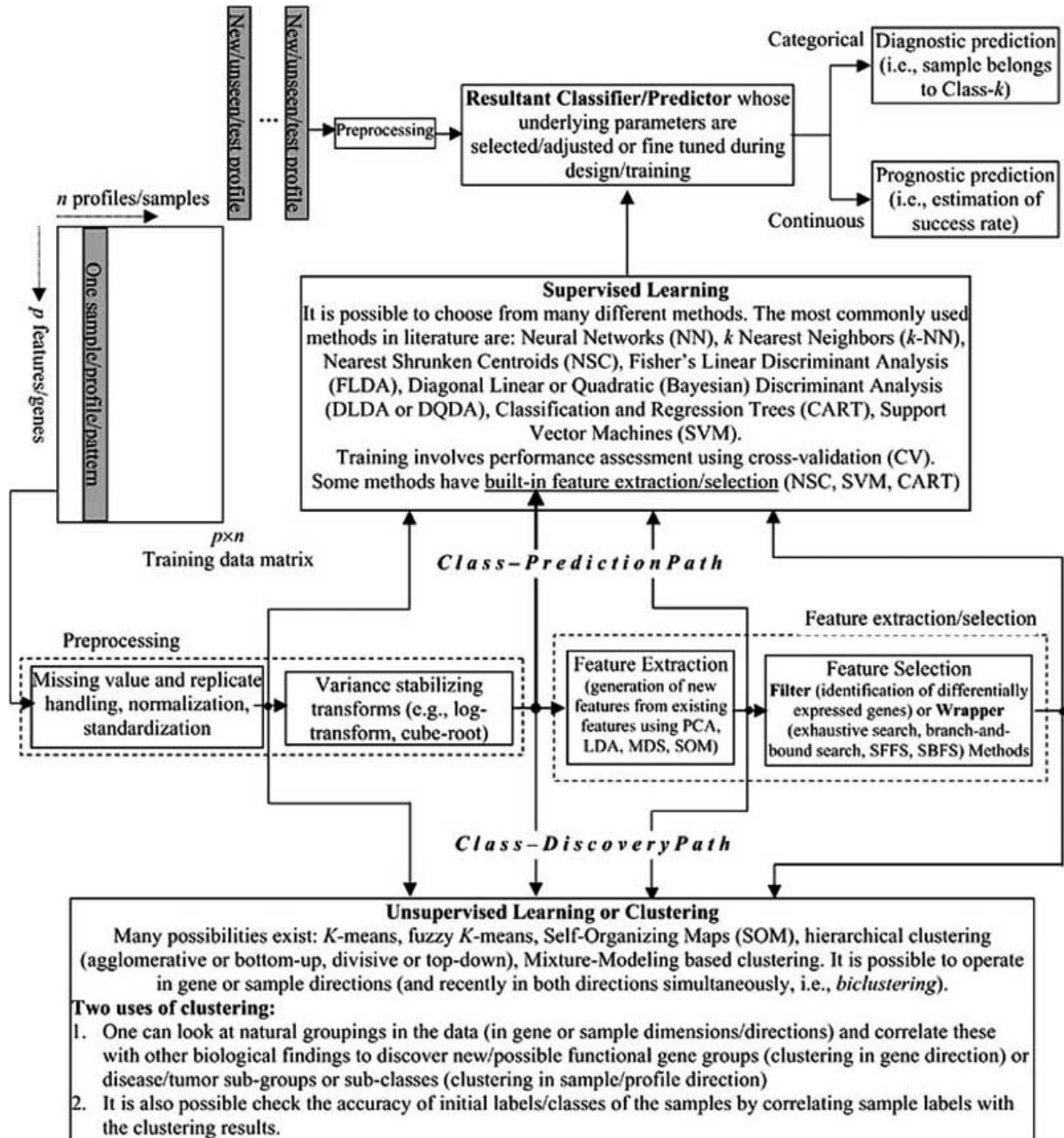


Fig. (2). Global picture for gene expression data analysis geared towards class-prediction or class-discovery.

2. A CRITICAL REVIEW OF PATTERN CLASSIFICATION METHODS APPLIED TO GENE EXPRESSION DATA

2.1. Basic Concepts

In a *class prediction* study, the upward path in Fig. 2, we basically design a class-predictor using already available *training* or *learning* samples (i.e., gene expression profiles) from different diagnostic classes. (For brevity, we will refer to *class-predictor* as *classifier* from here on.) Given the training samples representing different classes and the particular choice of the classifier design approach, first the

classifier is trained (i.e., estimate or adjust the parameters of the underlying approach) and then the classifier is used to predict the class membership (i.e., diagnostic class) of new or unseen samples. The design of a classifier involves development of decision rules, mathematical formulas, or models, using a particular classifier design strategy, based on the information available in the training set. In this sense, the classifier can be thought of as an artificial intelligence device that has the capability of making diagnostic or prognostic predictions. Given a new pattern or profile that the classifier did not see during training, the classifier uses some certain features determined during the feature extraction/selection

phase. The classifier processes the information from the relevant features in a certain way, which is learned or optimized during the training phase, and outputs its class prediction or decision. An excellent example of a class-prediction study can be seen in [21].

Hearing about the availability of such mathematical tools may sound exciting to the physicians who are challenged with complex diseases/conditions such as cancer. However, due to the immature state of the microarray technology and the performance limitations of classifiers trained using only a few samples, such a diagnostic and/or prognostic class predictor can only be considered as a research tool that could offer limited help to the health care professionals. Another inherent limitation associated with this type of study is that existing classes in the training data may be heterogeneous, i.e., there may be sub-classes within our predefined classes whose molecular and clinical course may be quite different. On the other hand, it will be seen in literature that accurate prediction performance of the designed/proposed classifiers are used as a proof to show the importance/significance of the features (genes) selected during the *feature selection* phase of the classifier design. Actually, in many cases, the results of the intermediate steps involved during classifier design (e.g., the identification of differentially expressed genes in the context of feature selection) may be as interesting as the resultant classifier itself.

2.2. Preprocessing

The term “gene expression” that has been used is somewhat fuzzy, as each different microarray technology offers a different way of measuring gene expression. As long as the samples that constitute the training dataset are obtained using the same technology and possibly preprocessed in the same manner, we do not need to be concerned about this particular issue.

The preprocessing of gene expression data, before carrying out a supervised or unsupervised classification study, may be needed for several reasons. Sometimes, due to technological problems or mishandling of the microarrays, expression values for some genes cannot accurately be measured, in which case the problem of missing data arises. Classification methods do not generally have the capability or provision to handle missing data (one exception is the classification tree). Therefore, the missing values need to be filled in or *imputed* with some reasonable estimates before proceeding with the classification study. Not doing so results in discarding genes (rows) with missing entries in the gene expression data matrix, therefore precious training data may be seriously reduced and consequently its ability to represent the investigated biological situations/conditions may diminish. Given a row/gene with some missing data, if the number of the missing values is relatively low, compared to the number problem-free gene expression measurements from remaining profiles, corresponding gene's data can be salvaged by *imputation*, i.e. by assigning some reasonable values to missing data [40-42].

Another key preprocessing step is the normalization or the method by which expression levels are made comparable. When gene expression measurements from different arrays are put together for a class prediction or discovery study, care must be taken, as conditions (target amount, labeling

efficiency, scanning efficiency, etc.) producing each array are different. A common approach to normalization is global normalization, where averages of the expression distributions (expression levels for all genes within an array) across different arrays are equalized. The rationale behind this approach is that while genes can be differentially expressed, the amount of transcription is essentially similar across samples. However, depending on the biological conditions compared, this assumption may be violated. A remedy is to use a set of common housekeeping genes on the arrays to facilitate normalization. Housekeeping genes are ubiquitously expressed across different biological conditions in a relatively stable manner. As opposed to normalizing to the average gene of the entire array, this form of normalization uses the average of the housekeeping genes. Normalization is a vast topic by itself; as such, its detailed discussion is beyond the focus of this review. It suffices to saying that different normalization techniques are used for spotted cDNA microarrays [43-54] and Affymetrix high density oligonucleotide arrays [51-54]. For a discussion of possible effects of normalization on gene expression data analysis see [55-57].

All of the classifier design and/or clustering algorithms use some distance or similarity measures to determine how close the samples or genes are to each other. Typically used distance and similarity metrics include the Euclidean distance and the Pearson's correlation. The Pearson's correlation is immune to shifts and scalings in the patterns whose similarity is to be measured. However, if the Euclidean distance is to be used, it is necessary to *standardize* the rows or columns, depending on the application, to make them have zero mean and unit variance. For instance, if the features are not standardized, nearest-neighbor class-predictors will overweigh those features that have larger variances. This may cause the classifier to ignore critical information from genes with low expression measures but strong interactions. Another preprocessing method is *scaling*. Neural network classifiers generally train better when the features values are small. Hence the features can be scaled to a specified range, for example -1 to $+1$. If a class-predictor design method preprocesses the training data, the identical preprocessing must be applied to the future data (i.e., new or unseen samples) as well.

Another relatively trivial preprocessing step is the handling of repeated gene expression values [58]. To collapse them into a single value, one can average them or take the median (if there are outliers), depending on the distribution of the repeated values. Literature on gene-expression data analysis also indicates that normalizing and transforming gene expression data using variance stabilizing transforms such as logarithm and cubic-root may help the classification algorithms to model the underlying structure in the training data more easily/accurately [59-63].

2.3. Classifier Design Methods

There are a plethora of classifier design approaches that we can select from. First, will be briefly discussed the ones that are most frequently used in gene expression profile classification. For a detailed description/review of each of these methods the reader is referred to [39, 64, 65]. Then, a comparison in Table 1 will be presented, to concisely

summarize different characteristics of supervised learning or class-prediction methods. It will be seen that some approaches can identify a few important features/genes and use the information from those only to do the prediction, yet some approaches can report the prediction strength in terms of class posterior probabilities.

The most straightforward classifier design approach is based on the concept of similarity. In this approach, the distance between the test patterns whose class is to be decided and the known representatives or prototypes of classes are measured. Given a training set and a similarity measure or metric, to decide for the class membership of a test sample, the k nearest neighbors (k -NN) find the class membership of the k closest samples in the training set and takes a majority vote. The 1-NN classifier that assigns the test samples to the class of nearest observation in the training set is often used as a benchmark for other classifiers, since it always offers reasonable classification performance [39]. Typically used similarity metrics for k -NN classifiers include the Euclidean distance and Pearson's correlation. The number of neighbors k can be chosen by a leave-one-out cross-validated (LOOCV, will be discussed in Section 2.4) estimate of the error rate [19, 39]. Example applications of k -NN on gene expression profiles can be found in [19, 20, 22, 66].

In the nearest mean classifier, the prototypes are the class means/centers or centroids. Tibshirani *et al.* [21] suggested an enhancement for the nearest centroid classifier, called Nearest Shrunken Centroids (NSC). (The NSC is also referred to as PAM, Prediction Analysis of Microarrays, due to the name of the associated paper and software.) In NSC, weak components of the class-centroids are shrunk or deleted *via* soft-thresholding. The classification accuracy (expressed in terms of training, test, and cross validation error rates) and the number of present (or undeleted) genes are plotted against a parameter called *delta* that adjusts the amount of shrinkage and an optimal value for delta is selected by examining the error rates. Shrinkage eliminates the information that does not contribute towards class prediction, i.e., noise. The contribution or strength of each class centroid to the classification is measured by a t -statistics, where the numerator is the difference between individual class means and the overall mean and the denominator is the pooled estimate of standard deviation inflated by a fudge factor.

Another popular classifier design approach is based on Artificial Neural Networks (ANN or simply NN). NN consist of many interconnected processing elements, called neurons, resembling human brain's structure. Through different structures (varying number of layers and number of neurons per layer), linear or nonlinear transfer functions that the individual neurons use, and training paradigms during which the weights of the connections are adjusted or tuned, the NN can model/reveal complex relationships among inputs (patterns) and outputs (class-decisions), exemplified or embedded in the training data. Although there is still considerable skepticism about NN among statisticians, NN have been successfully applied in a broad category of class-prediction problems. An example of NN in gene expression profile classification can be seen in Khan *et al.*'s study [67]. Jain *et al.* [39] discuss the parallelism that exist among

different NN and statistical pattern classification approaches in detail and then move on to say: "Despite these similarities, neural networks do offer several advantages such as unified approaches for feature extraction and classification and flexible procedures for finding good, moderately nonlinear solutions."

Other popular classifier design approaches include Fisher's Linear Discriminant Analysis (FLDA). The FLDA is both a class-predictor design and a feature extraction/selection approach, or expressed differently, FLDA is a classifier design approach with built-in feature extraction/selection capability. A linear discriminant function is nothing but a special linear combination of the values of all the features that are used in classifier design. In FLDA, to achieve interclass separation, an optimal set of basis functions is found as the eigenvectors of the product of the inverse of the within-class scatter matrix and the between-class scatter matrix [19, 39, 64]. Since $K \ll p$, K being the number of distinct classes in the training data, there are at most $K-1$ discriminant axes corresponding to $K-1$ non-zero eigenvalues. Once the orthogonal basis vectors representing optimal projections are obtained, the class centers estimated from the training data are projected onto this vector space. Then, to predict the class of a test sample, the sample is also projected onto this space, which is optimal for class separation, and the Euclidean distances between the projected pattern and class centers are compared (the pattern is assigned to the class whose center is closest). The performances of FLDA and other discriminant analysis methods in gene expression profile classification are thoroughly studied by Dudoit *et al.* [19]. The major difficulty with this popular approach is that even though it uses only $K-1$ features in the projection space, while predicting the class of a new/test sample we need the measurements for all the features/genes so that it can be projected on the optimal discriminant axes. Further, Dudoit *et al.*'s study [19] shows that FLDA may not perform well if the number of genes was large relative to the number of samples. This performance degradation is due to the correlation among the genes/features that the FLDA takes into account while finding the optimal projections. If there are many genes in the classifier, there will be too many correlations to be estimated. In small sample size situations, those correlations cannot be estimated accurately (i.e., estimated parameter will have high variances due to low n), which makes the method unstable.

In order to introduce other major classifier design approaches such as DLDA and DQDA that are frequently used in gene expression profile classification, we will briefly review the Bayesian decision theory. In this model based setting, the class conditional densities are assumed to have multivariate normal densities typically. Given training samples we can estimate the class-conditional densities empirically using the maximum likelihood approach [39]. Let \mathbf{x} denote an observation in a p -dimensional feature space (i.e., $\mathbf{x} \in \mathbf{R}^p$) with a probability density of $f(\mathbf{x})$ and $k = 1, 2, \dots, K$ be the class index. If class prior probabilities $P_k = P(\mathbf{x} \in \text{Class-}k)$ and the class conditional densities $f(\mathbf{x}; \mathbf{x} \in \text{Class-}k)$ are known, according to Bayes formula, the posterior class probabilities are:

$$f(\mathbf{x} \in \text{Class-}k; \mathbf{x}) = f(\mathbf{x}; \mathbf{x} \in \text{Class-}k)P_k / f(\mathbf{x}). \quad (1)$$

The $f(\mathbf{x})$ in the denominator of Eq. (1) is a scaling factor that ensures the probabilities, when summed over k , added up to 1, therefore it can safely be ignored while comparing the posterior probabilities. Assuming equal risk or cost of misclassification for different classes, the Bayesian or minimum-error-rate classification decision rule simply assigns an observation to the class that provides the highest posterior probability:

$$\mathbf{x} \in \arg \max_k f(\mathbf{x} \in \text{Class-}k; \mathbf{x}).$$

Because of this feature, the Bayesian approach is also known as the Maximum A Posteriori Probability (MAP) approach. The class posterior probabilities may be used to assess the prediction strength. The following discriminant functions (DF) which are based on the logarithm transform of posterior probabilities, can be used (i.e., maximized over k) to assign samples to classes:

$$g_k(\mathbf{x}) = \ln f(\mathbf{x}; \mathbf{x} \in \text{Class-}k) P_k = \ln f(\mathbf{x}; \mathbf{x} \in \text{Class-}k) + \ln P_k.$$

It is noted that the prior probabilities bias the decisions in favor of the more likely classes. If the class conditional densities $f(\mathbf{x}; \mathbf{x} \in \text{Class-}k)$ are multivariate normal, that is

$$f(\mathbf{x}; \mathbf{x} \in \text{Class-}k) = N(\mathbf{x}; \mu_k, \Sigma_k) = (2\pi)^{-p/2} |\Sigma_k|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{x} - \mu_k)^T \Sigma_k^{-1} (\mathbf{x} - \mu_k)\right\}$$

where μ_k is the $p \times 1$ mean vector, Σ_k is the $p \times p$ variance-covariance matrix, and the $|\Sigma_k|$ is its determinant, the DF can be simplified as:

$$g_k(\mathbf{x}) = -\frac{p}{2} \ln(2\pi) - \frac{1}{2} \ln |\Sigma_k| - \frac{1}{2} \{(\mathbf{x} - \mu_k)^T \Sigma_k^{-1} (\mathbf{x} - \mu_k)\} + \ln P_k.$$

The first terms in the DF are independent of the class index k , therefore, can be dropped. If we further assume that the K classes have the same covariance matrix, i.e., $\Sigma_k = \Sigma$, $k = 1, 2, \dots, K$, DF are reduced to:

$$g_k(\mathbf{x}) = -\frac{1}{2} \{(\mathbf{x} - \mu_k)^T \Sigma^{-1} (\mathbf{x} - \mu_k)\} + \ln P_k. \quad (2)$$

The assumption that each class has the same covariance matrix makes the classifier *linear*, as the decision boundary will be a hyper-plane in the p -dimensional space. If one assumes different covariance matrices for different classes, the classifier will have *quadratic* decision boundary.

We can expand the quadratic from $(\mathbf{x} - \mu_k)^T \Sigma^{-1} (\mathbf{x} - \mu_k)$ in Eq. (2) as:

$$(\mathbf{x} - \mu_k)^T \Sigma^{-1} (\mathbf{x} - \mu_k) = \mathbf{x}^T \Sigma^{-1} \mathbf{x} - \mathbf{x}^T \Sigma^{-1} \mu_k - \mu_k^T \Sigma^{-1} \mathbf{x} + \mu_k^T \Sigma^{-1} \mu_k.$$

Since $\mathbf{x}^T \Sigma^{-1} \mu_k$ is scalar and Σ_k^{-1} is symmetric, $\mathbf{x}^T \Sigma^{-1} \mu_k = (\mu_k^T \Sigma^{-1} \mathbf{x})$, hence the quadratic form reduces to:

$$(\mathbf{x} - \mu_k)^T \Sigma^{-1} (\mathbf{x} - \mu_k) = \mathbf{x}^T \Sigma^{-1} \mathbf{x} - 2 \mu_k^T \Sigma^{-1} \mathbf{x} + \mu_k^T \Sigma^{-1} \mu_k. \quad (3)$$

The first term in Eq. (3) can be dropped, as it is independent of class index:

$$g_k(\mathbf{x}) = \mu_k^T \Sigma^{-1} \mathbf{x} - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \ln P_k. \quad (4)$$

Here in Eq. (4), it is observed that the DF are simply some special linear combinations of gene expression values

plus a bias term that depends on the class centers and prior probabilities. For any given data point \mathbf{x} , the K DFs given by Eq. (4) are calculated and \mathbf{x} is assigned to the class with the maximum DF. Further details of the Bayesian Classification theory, including a discussion of the case where different classes have different covariance matrices, along with excellent pictorial representations of the corresponding decision regions can be found in Duda *et al.* representation [64]. The class means μ_k s and the covariance matrices Σ_k can be estimated from the samples as:

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{j \in \text{Class-}k} \mathbf{x}_j, \quad \hat{\Sigma}_k = \frac{1}{(n_k - 1)} \sum_{j \in \text{Class-}k} (\mathbf{x}_j - \hat{\mu}_k)(\mathbf{x}_j - \hat{\mu}_k)^T, \quad k = 1, 2, \dots, K$$

where n_k is the number of samples that belong to the k -th class. For the constant covariance matrix case we can use the pooled estimate of the covariance:

$$\hat{\Sigma} = \frac{1}{(n - K)} \sum_{k=1}^K (n_k - 1) \hat{\Sigma}_k.$$

A further simplification in this approach can be achieved if it is assumed that the covariance matrices are diagonal, that is, if the features/genes are independent. Thus, for linear- and quadratic-Bayesian classifiers, we respectively have $\Sigma = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_p^2)$ and $\Sigma_k = \text{diag}(\sigma_{k1}^2, \sigma_{k2}^2, \dots, \sigma_{kp}^2)$, $k = 1, 2, \dots, K$. Dudoit *et al.* [19] called these two classifier design approaches as Diagonal Linear and Quadratic Discriminant Analysis (DLDA and DQDA). These approaches are also known as Naive Bayesian, due to the independence assumption. Even though such an assumption is not realistic (considering the biological facts, i.e., co-regulation of the genes), due to methodological reasons (curse of dimensionality), it may yield better classification performance especially in small sample size situations.

The last two classifier design methods that will be covered are Classification and Regression Trees (CART) and Support Vector Machines (SVM). In CART approach, due to Breiman *et al.* [68], using node impurity criteria such as entropy (information content) and Gini's index of diversity [65], some important features are selected and binary splits are formed on those features repeatedly. Each terminal feature subset is associated with a class label. Dudoit *et al.* [19] identified three main aspects for tree construction: selection of splits, decision to declare a node terminal or to continue splitting, and assignment of each terminal node to a class. Depending on how these topics are treated, many variations of tree fitting are possible. Since the decisions/splits at nodes are binary, decision boundaries are parallel to the feature axes, as such, they are intrinsically suboptimal [39]. In CART design, pruning of decision tree is often performed to estimate the "best" tree size, i.e. the number of terminal nodes. The "best" here refers to a compromise between the complexity and the performance expressed by the classification error. The main advantage of decision trees is that understanding of their operation is very easy. We can trace the tree from top to bottom, by following "if...then" rules to see how the predictions/decisions are made. Because of their high transparency and ease of construction through freely available software, use of CART in class-prediction problems has become widespread

recently. Like the k -NN classifier, CART is also frequently used as a benchmark for other classifiers.

The SVM approach is one of the state of the art classifier design approaches. In contrast to parametric classifier design approaches that try to model class-conditional densities and construct decision based on these estimated densities, a totally different way of classifier design strategy is implemented in the SVM. This approach, due to Vapnik [69], is inspired by his philosophy: “if you possess a restricted amount of information for solving a problem, try to solve it directly and never solve a more general problem as an intermediate step. It is possible that the available information is sufficient for a direct solution but is insufficient for solving a more general intermediate problem.”

The SVM is basically a binary (two-class) classifier, however, it is possible to extend it for multi-class/category cases by solving several binary problems simultaneously and combining the decisions. A comprehensive discussion of multi-category SVM approaches applied to gene expression profile classification can be seen in Statnikov *et al.*'s study [66]. In SVM, the complexity of the classifier is based on the number of support vectors rather than the dimensionality of the feature space and this makes the algorithm less prone to over-fitting. The samples in the original feature space are projected onto a higher dimensional feature space where they can be separated by a maximal margin hyper-plane. The support vectors are the projected samples that determine the boundaries of the maximal margin separating hyper-plane. Once we the support vectors are found, new/test samples can be classified using them. The key idea in SVM implementation is to use kernel functions (linear, polynomial, radial basis function, etc.) in the original feature space that corresponds to the inner products in the projection (higher dimensional) space, where maximal linear separation will be achieved.

An important issue in classifier design is the feature selection. This refers to the idea that instead of using all available features, it may be better, for several reasons, to use a limited number of prominent features that reflect the class differences more clearly. Sometimes classifiers cannot even be designed when the number of features is larger than the number of samples, e.g., DLDA and DQDA. When there are too many features present, classifier/classification performance may degrade, i.e., classification error may increase or reach peak in some cases. This phenomenon is known as the *curse of dimensionality* leading to *peaking phenomenon* (see [39] for a demonstration of this phenomenon). Furthermore, the results of the feature selection study, i.e., the selected features (genes in our case) may be important by themselves, as they may lead to the discovery of disease/condition markers. Therefore, feature selection is another key area of research [70].

Another important issue in classifier/predictor design is the performance assessment, i.e., the estimation or measurement of prediction accuracy. The fundamental problem here is that it cannot be made sure that the given (limited number of) training samples are good representatives of their respective classes. In the case of gene expression profile classification, sample size is very low indeed. The prediction performance is sometimes expressed

using the notion of “ability to generalize.” As the name implies, *generalizability* refers to the prediction accuracy on new/unseen samples. Given a training dataset, we can train the classifier to recognize the samples in the training set, however, if the classifier learns more than the general characteristics of the samples, it may perform poorly on new/unseen samples. Extensive or *over-training* (a.k.a. *over-fitting*) may cause the classifier to model/learn even the inherent noise in the training data, which in turn may degrade the prediction performance on new samples. Therefore, we want the classifier learn just “as much as necessary” from the data so that it can also predict class membership of unseen samples sufficiently well. The same problem arises when a classifier has too many parameters, i.e. over parameterized. In such a case, the capacity of the classifier to classify or recognize different objects may be larger, however, its generalizability performance may be inferior.

Table 1 lists some important properties of the classifier design approaches that have been discussed so far. The comparison of computational requirements of different approaches needs a few words of caution. Assessment of classifier performance (i.e., classification error rate) is a crucial step in classifier design. This typically involves estimation of training and test error rates. As the training error is an optimistically biased estimator of true error rate, test error rate which is typically obtained through cross validation (CV) or bootstrap methods is used for classifier performance estimation. (This issue is discussed in detail in Section 2.6.) Depending on how CV or bootstrapping is done, computational requirements of classifier design approaches may change significantly. However, assuming that the method for the estimation of classification error rate is fixed, different classifiers can still be compared in terms of computational complexity. The comparison included in Table 1 is just meant to give some idea about the mathematical complexity of different classifiers. As there are many parameters that may change the computational requirements of different methods, making an accurate comparison is difficult.

2.4. Curse of Dimensionality

Generally speaking, the number of samples (n) must be larger than the number of features (p) for good classification performance. However, gene expression profiles consist of expression values for several thousand genes and considering the fact that only a few dozen of samples or profiles are available as training data, we are faced with the situation referred to as *curse of dimensionality* in the pattern recognition terminology [39]. In DLDA and DQDA type of classifiers, the variance-covariance matrix (estimated using the training samples) needs to be inverted to obtain the class-discriminant functions. When there are too many features and too few samples or observations, i.e., n is much smaller than p , the estimated variance-covariance matrix is singular, i.e., its inverse is not defined [64]. In order to cope with this problem, one can either use a dimension reduction technique to identify and select more important genes with which a well performing classifier can be designed. Alternatively, one can modify the estimated singular variance-covariance matrix by adding a constant times the identity matrix for instance, as in the case of ridge-regression, so that it

Table 1. Basic Properties of Classification Methods Applied to Gene Expression Data for Class-Prediction Purpose

Method	Important Properties	Comput. Cost	Transparency	Built-in FS?	Can report PS?	Generalizability
<i>k</i> -NN	Based on simple concept of similarity, as such metric dependent. Once metric is chosen, implementation independent. Very robust, frequently used as a benchmark for other classifiers.	Low	Low	No	No	Good
SVM	The complexity of the classifier is based on the number of support vectors rather than the dimensionality of the feature space. This makes the algorithm less prone to over-fitting.	High	Low	Yes	No	Good
CART	Decisions/splits at nodes are binary, hence decision boundaries are parallel to the feature axes, as such they are intrinsically suboptimal. CART is also frequently used as a benchmark for other classifiers.	Low	High	Yes	No	Can easily be over-trained (pruning may alleviate the problem)
NSC (PAM)	A variation of the nearest-mean classifier.	Low	High	Yes	Yes	Good
Neural Networks	Not transparent. Black box approach.	High	Low	Yes (MLP)	Yes (MLP)	Can easily be over-trained
FLDA	Collapses/projects all the features onto optimal axes, on which class separation (defined by BCV/WCV) is maximum	Low	Low	Yes	Yes	Bad, if n is too small compared to p , i.e., for $n \ll p$
DLDA	Assumes that the features are independent/uncorrelated. Very sensitive to low n . All the classes assumed to have the same covariance matrix, hence the decision boundary is a hyper-plane. Cannot be designed when $p > n$, so the initial p should be reduced using FS, before this approach can be applied	Low	Low	No	Yes	Bad, if n is too small compared to p , i.e., for $n \ll p$
DQDA	It differs from DLDA only in that different classes assumed to have different covariance matrices, hence the decision surface is a hyper-quadratic.	Low	Low	No	Yes	Bad, if n is too small compared to p , i.e., for $n \ll p$

Comput. Cost is the computational cost in terms of memory and processing time requirements. The transparency refers to ease of understanding of how the classifier processes data to reach its decisions, i.e., which features were effective in the predictions. The FS and PS respectively refer to feature/gene selection and prediction strength. The generalizability expresses the sensitivity of the classifier to over-training. (BCV: Between Class Variance, WCV: Within Class Variance, MLP: Multi-Layer Perceptron, n : Sample size, p : number of features).

becomes non-singular. Such methods are known as *regularization* techniques [71].

As discussed previously, in DLDA and DQDA approaches, it is assumed that the features are independent, hence adding a new feature will always reduce the probability of error or misclassification rate, as long as the added feature has a different class mean [64]. However, in practice, features are correlated, hence adding new features may not improve the classification accuracy. This remark is valid for not only Bayesian classifiers but also for other types of classifier. Additionally, for DLDA and DQDA classifiers, the class conditional densities are estimated from the training samples and plugged in for true parameters. For a fixed sample size, increasing the number of features increases the number of parameters (i.e., mean vectors and variance-covariance matrices in the case of normal class-conditional densities) to be estimated from the samples. This leads to higher estimation variance or less reliability for the estimated parameters. As a result, classification performance may decline by increasing the number of features.

The discussion of this important issue is concluded by the following remarks from Jain *et al.* [39]: “While an exact relationship between the probability of misclassification, the number of training samples, the number of features and the true parameters of the class-conditional densities is very difficult to establish, some guidelines have been suggested regarding the ratio of the sample size to dimensionality. It is

generally accepted that using ten times as many training samples per class as the number of features, i.e., $(n/K)/p > 10$ where K is the number of classes, is a good practice to follow in classifier design. The more complex the classifier, the larger should the ratio of sample size to dimensionality be to avoid the curse of dimensionality.” This requirement is somewhat too restrictive; later on in their review, while discussing feature selection, Jain *et al.* [39] suggest that $(n/K)/p > 5$ can also be satisfactory. Therefore, if we have for instance 100 gene expression profiles for a two-class (e.g., healthy versus diseased tissues) classification problem, we should consider finding at most 10 important genes, to design a classifier with acceptable generalizability performance.

2.5. Feature Selection

In order to avoid the peaking phenomenon caused by the curse of dimensionality, one can select from the available features. As depicted in Fig. 2, the feature selection step can be preceded by feature extraction by which new or more useful can be generated. The idea in feature selection is to find the ones with better discrimination ability and design a classifier using those features only. Feature selection will also reduce the cost (i.e., the memory and computational requirements) associated with classifier design and more importantly, the cost of making predictions, assuming that the class-prediction study leads to the development of a diagnostic test based on gene expression profiling. Another

motivation for feature selection is related to Watanabe's Ugly Duckling Theorem [39, 64], which states that two patterns can be made arbitrarily close/similar to each other by adding extra features. In other words, having more features increases the chances of two arbitrary patterns being similar.

In Jain *et al.* [39] feature selection and feature extraction are treated separately under the heading of "dimension reduction." However, following the convention in bioinformatics literature, these topics will be covered under the title of feature selection. It has also been observed that in literature, biostatisticians and bioinformaticians do not strictly follow the already established pattern recognition terminology. For instance, Principal Component Analysis (PCA) is a feature extraction (or new feature generation) algorithm, however, many papers/studies in bioinformatics literature refer to PCA as a feature selection technique. The PCA just finds/suggests new features which are some optimal combinations of existing features. The optimality is in the sense that new features better account for the variation in the data. The PCA does not use the already available class membership information for the samples/observations in training set, therefore, it is an unsupervised method. FLDA, on the other hand, is a supervised feature extraction algorithm. For a K -class problem, it finds at most $K-1$ new discriminant axes maximizing the ratio between the inter-class and the intra-class variances. Duda *et al.* [64] make the following remark about PCA while comparing it to FLDA: "Although PCA finds components that are useful in representing data, there is no reason to assume that these components must be useful discriminating between data in different classes. If we pool all of the samples, the directions that are discarded by the PCA might be exactly the directions that are needed for distinguishing between classes. For example, if we had data for the printed uppercase letters O and Q, PCA might discover gross features that characterize Os and Qs, but might ignore the tail that distinguishes O from Q. Where PCA seeks directions that are efficient for representation, FLDA seeks directions that are efficient for discrimination."

Other feature extraction methods include kernel PCA, multidimensional scaling (MDS) and Self Organizing Maps (SOM). These are all examples of nonlinear feature extraction methods whose details can be found in [64]. Before we move on to the issue of feature selection, we should highlight the major practical difficulty associated with feature extraction algorithms. Feature extraction algorithms produce new features based on the values of all existing features. In the context of gene expression profiles, new features translate into *metagenes*, which are some particular linear combinations of the expression values for all the genes. Metagenes may correspond to the groups of genes functioning together, however, a classifier which uses such metagenes need the information from all the genes. Therefore, the use of metagenes is not practical for diagnostic test or biomarker development.

Ben-Dor *et al.* [22] proposed a clustering-based approach for classification. Their algorithm uses a threshold parameter to control the granularity of the resulting cluster structure where the similarity measure for the samples (i.e., the gene expression profiles) is the Pearson correlation. They

evaluated performance of their proposed method against large-margin classification methods (SVM and AdaBoost) and the nearest-neighbor method. Their results highlighted the importance of gene selection in improving the performance of their relatively simple clustering based classification approach.

Some classifiers like NSC, CART, and SVM have built-in feature selection and are relatively insensitive to the feature selection scheme. For instance, Statnikov *et al.* [66] showed that the performance of both SVM and other simpler approaches (k -NN, back-propagation and probabilistic neural networks [64]) improve *via* gene selection, however, the improvement is more emphasized in the case of non-SVM algorithms. In contrast, DLDA, DQDA, and k -NN classifiers do not perform feature selection, they use all the available features, without assessing their relevancy, in building the classifier.

If we exclude the implicit or built-in feature selection capability of certain classifiers, there are two fundamentally different approaches for gene selection: filtering (univariate) and wrapper (multivariate) methods. In filtering methods, the interactions among the genes are ignored and the class discrimination ability of each gene is considered separately using some ranking criteria. (Hence, these methods are also known as *one gene at a time* approaches.) For instance, if it is decided to use r (r is typically dictated by n/K , the number of training samples per class) genes in designing our classifier, we simply select the top r genes from the ranked list of p genes. The most frequently used ranking criteria include the ratio of between class variance (BCV) to within class variance (WCV), univariate parametric (nonparametric) test statistics such as t-test (Wilcoxon) for two classes or ANOVA (Kruskal-Wallis) for more than two classes [11, 72-76]. The filter approach is closely linked to the problem of finding *differentially expressed* genes. There is a vast literature on this topic. Pan [77] compared three parametric approaches, namely the t-test, a regression modeling approach, and a mixture model approach in his review, whereas Troyanskaya *et al.* [78], three nonparametric approaches: nonparametric t-test, Wilcoxon (or Mann-Whitney) rank sum test, and a heuristic method based on Pearson's correlation to a perfectly differentiating gene. The main distinction between filtering based gene selection and finding differentially expressed genes is that in gene selection we are not really concerned with the issues like multiple testing or false discovery rate [79], as the aim is just to rank the genes [80]. Therefore, neither the issue of finding the right cut-off on the calculated P -values using permutations [81], to obtain the set of differentially expressed genes, nor the issue of whether the assumptions behind the statistical test are satisfied or not is relevant.

The other major approach for feature selection is based on the gene subset search. In this so-called *wrapper* approach, given a set of p features, and a target reduced feature size, say r , the feature selection aims at finding the set of r features that minimize the classification error. In practice, r cannot be known in advance, however, we can have a rough idea about it, e.g., $r \ll n$, as dictated by the curse of dimensionality problem. A simple or direct approach for finding r best performing features out of p would involve trying all $p!/r!(p-r)!$ combinations in the

gene subset space with 2^p elements. For instance, if we had 100 genes/features and look for the best performing subset of genes with 10 elements, we would have to design about 1.731×10^{13} different classifiers and compare their prediction accuracy, using an exhaustive search. It is obvious that computational requirements of even such a simple problem are overwhelming, therefore many smart algorithms for searching the gene subset space have been proposed. Among these are Branch-and-Bound-Search (BBS), sequential forward (backward) selection, and sequential forward (backward) floating search. Only the exhaustive search and BBS guarantee an optimal subset [39]. For an excellent review of these methods, the reader is referred to [39, 70, 82].

Although the wrapper approaches have been extensively studied in the pattern classification literature, they are not widely used in the feature selection phase of gene expression profile classifier design. However, these approaches because of their multivariate nature (i.e., they consider the joint distribution of the features) can detect genes with weak main effects but possibly strong interactions. Inza *et al.* [83] and Levner [83] demonstrate the benefits of wrapper feature selection approaches in the gene expression profile and proteomic mass spectrometry analysis, respectively. Bo [84] concludes that evaluating combinations of genes when looking for differential expression between experiment classes reveal interesting information that will not be discovered otherwise. Further, to underline the merits of wrapper approaches, Jain *et al.* [39] in their review state that “In general good, larger feature sets do not necessarily include the good, small sets. As a result, the simple method of selecting just the best individual features may fail dramatically. It might still be useful, however, as a first step to select some individually good features in decreasing very large feature sets (e.g., hundreds of features). Further selection has to be done by more advanced methods that take feature dependencies into account. These operate either by evaluating growing feature sets (forward selection) or by evaluating shrinking feature sets (backward selection).”

The major dilemma coupled with the feature selection problem is the measurement of the performance (i.e., misclassification or error rate) of a class-predictor. The performance of a classifier cannot be assessed accurately when the training and/or test sample sizes are very low compared to the number of features. The importance of taking feature selection into account when assessing the performance of the classifier cannot be stressed enough. Feature selection is an integral step in designing a classifier, hence when using for example cross-validation to estimate classifier performance (i.e., generalization error), feature selection should be done not on the entire learning set, but separately for each cross-validation sample used to build the classifier. This point is also highlighted in Simon’s mini-review on tumor classification [80] using an excellent simulation study/example.

In summary, in order to reduce the complexity of classifier design problem and thus possibly avoid the curse of dimensionality, we can first generate new features (feature extraction through PCA or FLDA for instance) and then select a group/set of features with high class-discrimination ability, discarding the rest. If we identify a small number of

genes with which we can design a well-performing classifier, we can focus on those genes to better understand the molecular mechanisms behind the investigated conditions and possibly develop biomarkers.

2.6. Estimating Classifier Performance: Prediction Error

The issue of assessment of prediction error of a classifier also deserves much attention [85-87]. For parametric classifiers, given the class conditional densities, the probability of misclassification or Bayes error rate can also be used to quantify classifier performance, however, obtaining the analytic expression for this error rate is difficult in general. Therefore, empirical prediction/classification error remains to be the popular and practical performance measure. The empirical classification error is the ratio of wrong decisions to the total number of cases studied. The true error rate is statistically defined as the error rate of a classifier on an asymptotically large number of new cases that converge in the limit to the actual population distribution. During training, underlying parameters of a classifier are adjusted/estimated using the information contained in the training samples. The prediction accuracy can initially be evaluated by testing the classifier back on the training set and noting the resultant *training or resubstitution* (also known as *apparent*) error. This type of assessment of classifier performance, based on training error, is instrumental during the design phase. However, it may not be an accurate indicator of the final or overall performance of the classifier. As the interest is in employing the classifier in predicting diagnostic category of new or unseen samples, we also need to evaluate the generalizability performance of the classifier. Therefore, we should try to extrapolate the true error rate using the empirical error rates calculated from small samples.

Raudys and Jain [82] state that “The estimate of the classification error depends on the particular training and test samples used, so it is a random variable. One should, therefore, investigate the bias and the variance of the error rates estimates. In particular, one should always ask whether enough test samples were used to evaluate the classifier; were the test samples different from the training samples?” Given a classifier, if the number of incorrectly classified samples (out of total of n test samples) is τ , the maximum likelihood estimate P_e of P_e is given by $P_e = \tau/n$, with $E(P_e) = P_e$ and $Var(P_e) = P_e(1 - P_e)/n$ [39, 64]. Therefore, P_e is an unbiased and consistent estimator of P_e . Jain *et al.* [39] also demonstrate the importance of using confidence interval of margins of error while comparing classification errors of different classifiers on a sample case. An issue that is often overlooked while reporting classifier performance is that the confidence intervals are not given. Without confidence intervals, making statistically meaningful comparisons among classifiers is not possible. For instance, Lee *et al.* [20] in their comprehensive review of recent classifier design methods applied to gene expression data, merely reported the apparent classification error without the confidence intervals. Therefore, evaluating the performance of different classifier approaches based on their results would not be reliable. Furthermore, as will be seen in the Discussion and Conclusion Section, comparing classifier performances to identify the best method is a complicated issue.

If the training set contains too many samples with characteristics off the line with the population they represent (i.e., outliers), or excessive training is done so that the classifier learns or models even the inherent noise in the samples, the *generalizability* performance of the classifier will be poor. Therefore, while evaluating prediction accuracy of classification methods, it is important not to use the training error only [88]. In general, the training error rates tend to be biased optimistically, i.e., the true error rate is almost invariably higher than the training error rate. If there are plenty of training samples available, one can partition the overall training set into two sets and use one for training and the other for testing. However, the number of gene expression profile samples is generally too few to permit this. If we design the classifier based on a small training set, the generalizability performance of the classifier will be poor again. A common technique to assess classifier performance in such situations is to use m -fold cross-validation (CV). In this technique, the overall set of n training samples is randomly divided into m approximately equal size and balanced (i.e., the distribution of samples into different classes is similar) set of subsets. Then, each time one of these subsets is excluded from the overall training set and used as a test set (for the classifier that is trained using the remaining samples). This is repeated over the m sub-samples and the resultant test error rates are averaged to obtain the so-called m -fold CV error rate. However, m can also be set equal to n (size of the training set) in the case in which we have the Leave One-Out Cross-Validation (LOOCV). The case of $m=2$ is also known as the *holdout* method. For a comparison (in terms of bias, variance, and computational complexity) of different CV error estimates, the reader is referred to Table 9 in Jain *et al.*'s review [39].

If the classification error is estimated by randomly splitting the set of all available samples into two parts, as the size of training (testing) set increases (decreases), its bias decreases and variance increases [39, 64, 82, 89, 90]. Hence, the LOOCV error rate has low bias but high variance, as the size of the test set is 1. On the contrary, if m is set too low (i.e., when the size of the test set is larger), the m -fold CV error rate will have high bias and low variance. For a reasonable tradeoff between the variance and bias, typical choice for m in m -fold CV error estimation is 5 or 10. Further, 5 or 10-fold CV is computationally less demanding than LOOCV.

Another method for error estimation in small sample situations is the bootstrap resampling technique [90-92], which is an area of active research in applied statistics. In regular bootstrap error estimation, bootstrap or fake training sets of n samples are selected randomly *with replacement* from the original training set of n samples and 2-fold CV (i.e., equal size splits of train and test samples) error rate is calculated. The process of resampling and error rate calculation is repeated typically several hundred times and the average of resultant test error rates is an estimator for the true error rate of the classifier. A problem in this approach is that the samples being predicted in test set may also be in the bootstrap training set, in the case in which the classifier may predict it too well, leading to bias. Therefore, some enhancements for the bootstrap estimators are suggested. The two of such methods that have yielded better results for measuring classification error are the E0 and E632 bootstrap

[91]. For the E0 bootstrap estimator, test set is formed by the cases not found in the training set. It turns out that the average number of unique (non-repeated) and repeated samples in the training and test set are $0.632 \times n$ and $0.368 \times n$, respectively. The E632 bootstrap is the simple linear combination $0.368 \times \text{Resubstitution Error Rate} + 0.632 \times \text{E0}$. Both E0 and E632 are low variance estimators. For moderately sized sample sets, E0 is clearly biased pessimistically, since the classifier trains on only 63.29% of the cases on the average. However, E0 gives extremely accurate results when the true error rate is high. As the sample size grows, E632B is overly optimistic, but it is very accurate on small samples when the true error rate is relatively low. The bootstrap estimators are not always superior to LOOCV on small sample size situations. However, low E0 bootstrap estimate or repeated 2-fold CV is a stronger indicator of good classifier performance than LOOCV.

As stated earlier, if the classifier design includes a feature selection phase/step, the feature selection should be done on the CV training sets, and not on the whole training set. Leaving out feature selection from CV or other resampling (bootstrap) based performance assessment methods results in overly optimistic error rates. Simon [80] discusses many papers that appeared in prestigious bioinformatics journals where this issue was overlooked and misleadingly optimistic classifier performance results were reported. Ambroise and McLachlan [93] also highlight the same point while discussing the selection bias in gene selection on the basis of microarray gene-expression data.

3. A CRITICAL REVIEW OF CLUSTERING METHODS APPLIED TO GENE EXPRESSION DATA

Clustering, also known as unsupervised classification, looks for the natural groupings in a multidimensional dataset (e.g., gene expression profiles) based on a similarity or dissimilarity measure. Unlike class prediction, in clustering, classes are unknown and explored from the data itself. While clustering divides the data into similar groups, the classification assigns an observation (a gene or tumor tissue) to one of the already known groups. Clustering can be divided into two main categories; namely, partitional and hierarchical clustering. Given n samples with p genes each, the aim of the partitional clustering method is to partition the samples into K clusters so that the gene profiles in a sample group are more similar to each other than to gene profiles in different sample groups. A similar definition can be made if the aim is to cluster genes instead. The value of K is either pre-specified or can be estimated from gene expression profile dataset [39, 94, 95].

We can cluster genes into groups or cell lines (samples) based on gene expression. The latter can be instrumental in identification of new tumor classes while the former in identification of the groups of co-regulated genes. Every clustering method will produce clusters but the clustering methods do not guarantee that the discovered clusters are biologically meaningful. However, clustering can be very useful as an exploratory tool.

There are two popular partitional clustering strategies: square-error and mixture modeling. The sum of the squared Euclidian distances between the genes in a cluster and the

cluster center is called within-cluster variation. The sum of the within-cluster variations in a clustering scheme is used as a criterion in K -means clustering [64, 96, 97]. This clustering is also known as minimum variance partition. K -means clustering is computationally efficient and gives satisfactory results if the clusters are compact and well separated in the feature space. Incorporating fuzzy criterion into K -means clustering results in fuzzy K -means clustering (also known as fuzzy C -means, FCM) in which each data point has a degree of membership to each class [27, 98-100]. The concept of degree of membership in fuzzy clustering is similar to the posterior probability in a mixture modeling setting. By monitoring data points that have close membership values to existing classes, forming new clusters is possible; this is the major advantage of fuzzy K -means clustering over regular K -means clustering.

Yeung *et al.* [101] and McLachlan *et al.* [102] discussed the use of model-based gene expression profile clustering approaches for classification. They used Gaussian mixture models to cluster the samples and eventually tested the groupings in the samples against apriori known class information for the samples.

Mixture modeling based clustering assumes that each measurement comes from a distribution characterized by a probability density function that is a *mixture* of several (a finite number of) components. It is generally assumed that all the density components have the same functional form, multivariate normal, for instance. Finite mixture modeling is a widely used technique for probability density function estimation [103-106] and found significant applications in various biological problems. The mixture components (i.e., their underlying parameters such as mean and variance for the case of normal mixtures) and their weights can be estimated using the EM algorithm [107-109], which is an iterative method for optimizing the likelihood function in situations where there is some missing information (e.g., the class memberships of the data points). Typically, the K -means algorithm is used to initialize the EM to ensure that it will find a "good" local maximum [104]. This is often considered sufficient in practical applications [110]. Although, there are other ways of fitting or estimating mixture models, the EM algorithm is relatively easy to implement, does not require any gradient computation, and is numerically stable. Once the mixture components are estimated, posterior probability of belonging to different classes can be evaluated and the class-membership can be decided for each data point. Mixture modeling based algorithms perform superior compared to other methods such as K -means, when the data is coming from overlapping densities [39].

In the context of gene expression, Hierarchical clustering (HC) was first used by Eisen *et al.* [111] to visualize multidimensional cDNA microarray data. HC starts with each object (sample or gene) as a singleton cluster, and then at every level, the dissimilarity matrix, a matrix representing the pair-wise dissimilarity between clusters, is updated using a distance metric to form the next layer. The average-linkage method, which uses the average of the pair-wise distances between the members of two clusters, is a common one. Alizadeh *et al.* [17] used HC with centroid linkage and Pearson correlation based distance metrics on both gene and

sample axes. The end product of the HC is a tree or dendrogram. Dendrograms can be built in two different ways; bottom-up (agglomerative) or top-down (divisive). Table 2 shows the classical clustering methods applied to gene expression data along with their important properties, pros, and cons.

In an earlier study, Tavazoie *et al.* [112] used K -means clustering to group genes in synchronized *Saccharomyces cerevisiae* batch cultures with 15 time points, across two cell cycles. Following the clustering, they searched for regulons (sets of co-regulated genes) within each group. This earlier study demonstrated two things. First, K -means algorithm was able to group the genes into biologically valid clusters. Second, clustering algorithms in general could be used to discover co-regulated genes or gene networks.

Granzow *et al.* [24] compared K -means and fuzzy K -means clustering methods along with Kohonen networks, the growing cell structure networks and fuzzy Kohonen networks on the acute myeloid leukemia and acute lymphoblastic leukemia data set. Fuzzy Kohonen neural network produced the most homogeneous clusters with respect to the tumor classes. McLachlan *et al.* [113] used mixture-modeling based approach for reducing the gene dimension (i.e., number of genes) and clustering tissue samples. Before clustering using a mixture of factor analyzers, they used mixtures of t distributions, instead of normals, to cluster samples "to provide protection against atypical observations". They claim that the degree of freedom parameter in the t distribution provided an adaptive and robust modeling. In the second stage they cluster the reduced set of genes into a specified number of groups.

Ghosh and Chinnaiyan [95] also proposed a mixture modeling of gene expression data to cluster either genes or samples. They initialized the EM algorithm with HC agglomerative clustering, in which, instead of a dissimilarity metric such as average-linkage approach, the log-likelihood probabilistic criterion was used to form the clusters. The number of clusters was adaptively estimated using the Bayesian Information Criterion (BIC) of [114]. Due to the dimensionality problem, before fitting the mixture model, genes were grouped into $K = 1000$ clusters using K -means clustering. These clusters were the input to their model-based agglomerative HC clustering. During the clustering of samples, principle component analysis was used to reduce the gene dimension, then, the same mixture model was applied to the reduced data set. Although it is not explicitly made clear, it should be noted that the method of Ghosh and Chinnaiyan [95] is a hybrid method of K -means, agglomerative HC and mixture-model based clustering methods. In general, although K -means is not as favorable as its counterparts, due to its computational efficiency and acceptable results in many cases, it may be used to initialize the other more complicated clustering algorithms.

Recently, several new clustering algorithms are graph theory-based (such as CLICK [115] and CAST [116]) and density-based hierarchical approach (DHC) [117] has been introduced for clustering the gene expression data. The graph-theoretical clustering techniques are converting the problem of clustering a data set into finding a minimum cut or maximal cliques in the proximity graph where each data point corresponds to a vertex. CLICK (CLuster Identification

Table 2. Classical Clustering Methods Applied to Gene Expression Data

Method	Properties	Pros	Cons
<i>K</i> -means	Identifies clusters by minimizing the overall within-cluster variance.	Computationally very efficient. It can find hyper-spherical or hyper-ellipsoidal clusters. Due its practicality and reasonable results, is frequently used to initialize other more complicated clustering methods to speed up convergence.	<i>K</i> and initial cluster centers need to be specified. Clusters have to be compact and well separated.
Fuzzy <i>K</i> -means (Fuzzy <i>C</i> -means)	Similar to <i>K</i> -means but every object has a degree of membership to the <i>K</i> clusters.	The degree of membership information is helpful in identifying new clusters.	Computationally inefficient and may require additional parameters.
Mixture modeling	Objects are assumed to be drawn from the mixture of <i>K</i> distributions. Distribution parameters are estimated using the well-known EM algorithm.	Always converges to local minimum. Better than <i>K</i> -means when the classes are overlapping distributions. It has probabilistic measure of membership.	<i>K</i> and initialization is required.
Hierarchical clustering	Clusters using agglomerative or divisive approach. Uses a dissimilarity matrix to form the tree or dendrogram.	Computationally efficient.	Output is a dendrogram, not clusters per se. Different clusters can be found by cutting the dendrogram at different levels.

via Connectivity Kernels) [115] iteratively finds the minimum cut in the proximity graph and splits the data set recursively into a set of connected components from the minimum cut. CLICK assumes that the similarity values within clusters and between clusters are normally distributed. Similarly, CAST (Cluster Affinity Search Technique) [116] considers a probabilistic model in designing a graph theory-based clustering algorithm. The CAST algorithm searches the clusters one at a time. The relation between a data point and a cluster is determined by the affinity value. If it satisfies certain criterion (high affinity value/low affinity value), the elements are added/removed into/out of the current cluster. The process continues with each new cluster until all elements are assigned to a cluster.

Many clustering algorithms require the number of clusters *K* to be provided. The quality of the resulting clusters is highly dependent on the estimation of *K*. Too many clusters would complicate the result and would be hard to interpret; on the other hand, with too few clusters it may cause the loss of information and may mislead the final decision. Therefore, there have been some attempts to make an estimate of *K* [118]. However, Xu *et al.* [118] state “Constructive clustering algorithm can adaptively and dynamically adjust the number of clusters rather than using a pre-specified and fixed number.”

In this section, we went through the most common methods employed in clustering the gene expression data. For a comprehensive review of the clustering methods, the reader can refer to [118, 119]. Each clustering algorithm performs clustering based on different criteria and assumptions. Hence, the performance and ability of each algorithm would vary with different gene expression data sets. Therefore, it would not be accurate to claim the “best” in the context of the clustering algorithms. However, one can make some comparisons. For instance, if the number of clusters is known and the data set contains few outliers, *K*-means and SOM may perform better. On the other hand, if the dataset is very noisy or the number of clusters is not known, CAST or CLICK may be a better choice.

4. BICLUSTERING OF GENE EXPRESSION DATA

The recognition of patterns hidden in gene expression data offers a tremendous opportunity for an enhanced understanding of functional genomics. The clustering methods discussed in the previous section play a big role in microarray data analysis. Clustering techniques can be used to group either genes or conditions. The standard clustering methods, such as hierarchical clustering [111], *K*-means [120], or self-organizing maps (SOM) [121], assume that genes in a cluster behave similarly over all measured conditions, and the resulting clusters are exclusive and exhaustive.

The standard clustering methods may produce reliable results for microarray experiments performed on homogeneous conditions, which may be reasonable for a single, focused experiment. However, for large and heterogeneous gene expression data sets describing the transcriptional response to a variety of different experiment conditions, this assumption will become no longer appropriate. In addition, grouping of genes into disjoint clusters as is done in traditional clustering algorithms may preclude genes from involving in multiple biological functions or processes, which may not be an accurate representation of the biological system [122].

As Jiang *et al.* [119] states “It is well known in molecular biology that only a subset of the genes participates in any cellular process of interest and that any cellular process takes place only in 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 no cluster at all”. Hence, the expression levels of a subset of genes might only show coherency under a subset of conditions. Therefore, “global clustering” over all dimensions (conditions) may separate biologically related genes from each other and hence may not discover local expression patterns in the microarray data. Therefore, it is desirable to find local expression patterns in microarray gene expression data. Recently, biclustering algorithms have been proposed to cluster both genes and conditions simultaneously [123]. Hence, unlike traditional

clustering algorithms, biclustering can identify subset of genes that are similarly expressed over a subset of experimental conditions that better reflects the biological reality. This approach can overcome the shortcomings associated with the traditional clustering methods.

The biclustering algorithm which was first introduced by Hartigan [124], and termed direct clustering, tried to identify local sub-patterns or -matrices in an arbitrary data matrix; And it was first applied to gene expression data by Cheng and Church [123]. In the literature, several different names are used to refer to biclustering, such as co-clustering, subspace clustering, bi-dimensional clustering, two-way clustering, and block clustering. There is an extensive literature on biclustering algorithms. The biclustering methods differ in terms of 1) the characteristics of the bicluster it can find based on a homogeneity merit function (biclusters with constant values, or constant values on the rows or the columns of the data matrix or with coherent values on the rows and columns), 2) the way the bicluster structures are obtained (one bicluster or non-overlapping biclusters, or multiple overlapping biclusters), and 3) type of algorithm being used (Greedy iterative search, iterative row and columns clustering combination, divide and conquer, exhaustive bicluster enumeration, and distribution parameter identification using a given statistical model). This section will briefly present some representative biclustering algorithms applied to gene expression data. The reader is referred to [125] and the references therein for a extensive survey of biclustering methods.

The biclustering algorithms that will be described below are tested on the gene expression data sets obtained either from yeast cells [123, 126-128], or human cells (mostly from cancerous tissues) [123, 127, 129-131]. The algorithms' performances are tested in the areas of identification of co-regulated genes, gene functional annotation, and sample and/or tissue classification.

Cheng and Church [123] introduced the concept of bicluster to identify uniform sub-matrices corresponding to a set of genes showing coherence over a set of conditions. Mean squared residue scoring is proposed to quantify the uniformity of the sub-matrix, and a greedy algorithm is employed to identify large sub-matrices with high similarity scores. The algorithm is based on the deletion and addition of rows and columns to iteratively improve the score of each bicluster. The algorithm discovers one bicluster at a time. The discovered biclusters are then masked with random numbers to allow the identification of new clusters in subsequent runs. The process is continued until pre-specified number (K) of clusters has been identified. However, the limitation of this method is that due to masking of discovered biclusters with random numbers, the discovery of highly overlapping biclusters would not be possible.

Yang *et al.* [132, 133] introduced the algorithm called FLOC (FLexible Overlapped biClustering) addresses this limitation. FLOC performs simultaneous bicluster identification. It starts with K randomly selected sub-matrices. At each iteration, every row/column are added/removed into/out of the subspace clusters to lower the residue value. The residue of the bicluster is computed by the existing values of the data matrix. Hence, FLOC is robust

against missing values. It can also discover a set of possibly overlapping biclusters simultaneously.

The Coupled Two-Way Clustering (CTWC) [129, 130, 134] is initialized by clustering the genes and the conditions of the data matrix separately. A hierarchical clustering algorithm called Super-Paramagnetic Clustering algorithm (SPC) [135] is used to generate stable clusters of rows and columns. At each iteration, one stable row subset and one stable column subset are coupled and two-way clustering is then applied to all sub-matrices in the following iteration. All pairs of previously identified clusters are used to generate the sub-matrices in the following iteration. The CTWC dynamically maintains two lists of stable clusters (gene clusters in GL and sample clusters in SL). At each iteration, one gene subset from GL and one sample subset from SL coupled and clustered mutually. The newly generated stable clusters are added to GL and SL together with a pointer that identifies the parent pair to indicate the origin of the clusters. The iteration continues until no new clusters satisfying some criteria, such as stability or critical size, are identified. Any standard clustering method can be used within the framework of CTWC.

Tanay *et al.* [127] introduced SAMBA (Statistical-Algorithmic Method for Bicluster Analysis) method which combines graph theory with statistical data modeling. The algorithm performs simultaneous bicluster identification by using exhaustive bicluster enumeration. The gene expression data matrix is modeled as a weighted bipartite graph of genes and conditions that are connected with edges for significant expression changes. The method is based on a bicluster scoring scheme that uses a statistical model to assign weights to vertex pairs so that finding statistically significant biclusters corresponds to identifying heavy sub-graphs in the data. The heuristic search used to identify these sub-graphs is guaranteed to find the most significant biclusters. In a more recent study, Tanay *et al.* [136] extended the SAMBA framework to model diverse collection of genome-wide data sets including gene expression, protein interactions, phenotypic measurements or transcription factor binding site to identify modules.

The plaid model introduced by Lazzeroni and Owen [126] regards the gene expression data matrix as a linear function of multiple "layers" corresponding to its biclusters. The clustering process searches for the layers in the data set using the EM algorithm and uses Lagrange Multiplier to estimate the model parameters. New layers are added to the model one at a time. Suppose that first $K-1$ layers have been extracted, the K th layer is identified by minimizing the sum of squared errors. The biclustering process stops when the variance of expression levels within the current layer is smaller than a threshold. The plaid model can be seen as a generalization of additive model, i.e., the values in the matrix are considered as a sum of contributions from multiple biclusters in which they belong to (in the case of overlapping biclusters) [125].

Sheng *et al.* [131] introduced Gibbs sampling method to the biclustering of discretized microarray data. The biclustering problem is cast into a Bayesian framework and parameters for both row and column distributions are estimated by using Gibbs sampling. Their approach identifies one bicluster at a time. To detect multiple

biclusters, they mask the genes that belong to the previously found bicluster to allow for the identification of multiple biclusters. However, the masking scheme used here precludes overlaps in the gene content of the resulting biclusters. In addition, this approach requires the discretization of expression data into fixed number of bins. Wu *et al.* [137] applied the Gibbs sampling scheme to the biclustering of continuous gene expression data. They implemented the algorithm in the program called GEMS (Gene Expression Module Sampler).

5. DISCUSSION AND CONCLUSION

Due to recent advances in DNA microarray technology, it became possible to obtain gene expression profiles of samples from different disease/diagnostic classes. Several classification algorithms based on statistical analysis have been applied on these profiles, in an attempt to achieve accurate and automatic class prediction. Compared to conventional tissue classification techniques like visual inspection under light-microscope, advantages of this type of classification are many-fold. Firstly, these are quantitative and systematic approaches; secondly, recent studies report high class prediction accuracies using these methodologies, proving that the idea of class prediction based on gene expression profiles is realistic. Furthermore, some approaches can pinpoint relatively important genes in class prediction, which in turn may lead to the discovery of genes that are responsible for certain conditions, and also indicate the certainty of the class predictions in terms of class membership probabilities.

This review has discussed the class-prediction and discovery methods that are applied to gene expression data, along with the implications of the findings. Recently, several reviews about this topic have appeared in literature [19, 20, 66, 80]. However, the scope of most reviews is limited by the class-prediction. Here, it was attempted to present a unified approach that considers both class-prediction and class-discovery. A substantial part of this review was devoted to an overview of pattern classification/recognition methods and discussed important issues such as preprocessing of gene expression data, curse of dimensionality, feature extraction/selection, and measuring or estimating classifier performance. Although these concepts are relatively well understood among the technical people such as statisticians, electrical engineers and computer scientists, they are relatively new to biologists and bioinformaticians. As such, it was observed that there are still some misconceptions about the use of classifications methods. For instance, in most classifier design strategies, the gene/feature selection is an integral part of the classifier, as such it must be a part of the cross-validation process that is used to estimate the classifier prediction performance. Simon [80] discusses several studies that appeared in prestigious journals where this important issue is overlooked, and optimistically biased prediction performances were reported. Furthermore, we have also compared and summarized important properties such as generalizability (sensitivity to overtraining), built-in feature selection, ability to report prediction strength, and transparency (ease of understanding of the operation) of different approaches to provide a quick and concise reference.

The classifier design and clustering methods (in short classification methods) are relatively well established, however, the complexity of the problems rooted in the microarray technology hinders the applicability of the classification methods as diagnostic and prognostic predictors or class-discovery tools in medicine. Furthermore, the question of classification “which method is better” does not have a simple answer. In the Summary Section of Chapter 9 (Algorithm Independent Machine Learning) of their famous book on pattern classification, Duda *et al.* [64] make the following important remarks: “The No Free Lunch Theorem states that in the absence of no prior information about the problem there are no reasons to prefer one learning algorithm of classifier model over another. Given that a finite set of feature values are used to distinguish the patterns under consideration, the Ugly Duckling Theorem states that the number of predicates shared by any two patterns is constant and does not depend upon the choice of the two objects. Together, these theorems highlight the need for insight into proper features and matching algorithms to the data distribution: There is neither problem-independent “best” learning or pattern recognition system nor feature representation. In short, formal theory and algorithms taken alone are not enough, pattern classification is an empirical subject.” In this sense, in gene expression profile classification applications, we are in the middle of a dilemma. In one hand, we are trying to use classification/clustering approaches to get more insight about the underlying mechanisms (gene regulation) generating these patterns/profiles, and on the other hand, we need to know about such mechanisms in advance to select (come up with) the most-suitable class-predictor or clustering approach.

Biclustering has also been discussed as an emerging clustering method that sub-clusters the entries of the gene expression data matrix in both gene (row) and sample (column) directions simultaneously. Through biclustering, possible functional groups on the gene direction, new classes in the sample direction, and the links or coherence between them can be identified at the same time. In addition, biclustering is capable of discovering overlapping biclusters under different sets of conditions. There are a wide variety of biclustering algorithms introduced in the literature [125]. However, as Jiang *et al.* [119] state “there is no single best algorithm, which is the winner in every aspect”. Various approaches have been used to assess the quality or reliability of the biclusters, such as the value of the merit function which is an indication of the coherence in the resulting biclusters, statistical significance tests, and comparisons test against known solutions.

An interesting finding of years of pattern classification research is that there is no single approach that will entirely solve complex classification problems and some methods may perform better than others in some parts of the feature space [64]. Motivated by these facts, classifier combination has become a major topic of research recently. However, due to space limitations, the application or use of classifier combination in gene expression profile classification could not be covered in this review. This topic is extensively covered in [39].

Another area of active research that we could not cover is a new set of classification and/or clustering methods that may collectively be referred to as *knowledge based classification*. These methods try to integrate the information extracted from the gene expression data during clustering or classifier/predictor design studies with gene ontology [138-145]. As the ultimate goal of designing classifiers based on gene expression data is to understand the underlying molecular basis/mechanisms, these methods also seem to be promising.

REFERENCES

- [1] Shena M, Shalon D, Davis RW, Brown PO. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* **1995**; 270: 467-70.
- [2] Shalon D, Smith SJ, Brown PO. A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization. *Genome Res* **1996**; 6: 639-45.
- [3] Li C, Wong WH. Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection. *Proc Natl Acad Sci U S A* **2001**; 98: 31-36.
- [4] Irizarry RA, Hobbs B, Collin F, et al. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics* **2003**; 4: 249-64.
- [5] Nguyen DV, Arpat AB, Wang N, Carroll RJ. DNA microarray experiments: biological and technological aspects. *Biometrics* **2002**; 58: 701-17.
- [6] Ramaswamy S, Golub TR. DNA microarrays in clinical oncology. *J Clin Oncol* **2002**; 20: 1932-41.
- [7] Alizadeh AA, Ross DT, Perou CM, Rijn Mvd. Towards a novel classification of human malignancies based on gene expression patterns. *J Pathol* **2001**; 195: 41-52.
- [8] Nature-Genetics. The chipping forecast. *Vol 21 Supplement* **1999**: 1-60.
- [9] Nature-Genetics. The chipping forecast II. *Vol 32 Supplement* **2002**: 461-552.
- [10] Leung YF, Cavalieri D. Fundamentals of cDNA microarray data analysis. *Trends Genet* **2003**; 19: 649-59.
- [11] Reimers M. Statistical analysis of microarray data. *Addict Biol* **2005**; 10: 23-35.
- [12] Huber W, v.Heydebreck A, Vingron M. Analysis of microarray gene expression data., In: Balding DJ, Bishop M, Cannings C Eds, *Handbook of Statistical Genetics*. John Wiley & Sons, Chichester, 2003.
- [13] Peterson C, Ringner M. Analyzing tumor gene expression profiles. *Artif Intell Med* **2003**; 28: 59-74.
- [14] Lu Y, Han J. Cancer classification using gene expression data. *Inf Syst* **2003**; 28: 243-68.
- [15] Aittokallio T, Kurki M, Nevalainen O, Nikula T, West A, Lahesmaa R. Computational strategies for analyzing data in gene expression microarray experiments. *J Bioinform Comput Biol* **2003**; 1: 541-86.
- [16] Quackenbush J. Computational analysis of microarray data. *Nat Rev Genet* **2001**; 2: 418-27.
- [17] Alizadeh AA, Eisen MB, Davis RE, et al. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature* **2000**; 403: 503-11.
- [18] Golub TR, Slonim DK, Tamayo P, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **1999**; 286: 531-37.
- [19] Dudoit S, Fridlyand J, Speed TP. Comparison of discrimination methods for the classification of tumors using gene expression data. *J Am Stat Assoc* **2002**: 77-87.
- [20] Lee JW, Lee JB, Park M, Song SH. An extensive comparison of recent classification tools applied to microarray data. *Comput Stat Data Anal* **2005**; 48: 869-85.
- [21] Tibshirani R, Hastie T, Narasimhan B, Chu G. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc Natl Acad Sci USA* **2002**; 99: 6567-72.
- [22] Ben-Dor A, Bruhn L, Friedman N, Nachman I, Schummer M, Yakhini Z. Tissue classification with gene expression profiles. *J Comput Biol* **2000**; 7: 559-83.
- [23] Furey TS, Cristianini N, Duffy N, Bednarski DW, Schummer M, Haussler D. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* **2000**; 16: 906-14.
- [24] Granzow M, Berrar D, Dubitzky W, Schuster A, Azuaje FJ, Eils R. Tumor classification by gene expression profiling: comparison and validation of five clustering methods. *SIGBIO News* **2001**; 21: 16-22.
- [25] Halvorsen OJ, Oyan AM, Bo TH, et al. Gene expression profiles in prostate cancer: association with patient subgroups and tumour differentiation. *Int J Oncol* **2005**; 26: 329-36.
- [26] Jaeger J, Weichenhan D, Ivandic B, Spang R. Early Diagnostic Marker Panel Determination for Microarray Based Clinical Studies. *Stat Appl Genet Mol Biol* **2005**; 4: Article 9.
- [27] Wang J, Bo TH, Jonassen I, Myklebost O, Hovig E. Tumor classification and marker gene prediction by feature selection and fuzzy c-means clustering using microarray data. *BMC Bioinformatics* **2003**; 4: 60.
- [28] Zhu J, Hastie T. Classification of gene microarrays by penalized logistic regression. *Biostatistics* **2004**; 5: 427-43.
- [29] Brazma A, Hingamp P, Quackenbush J, et al. Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. *Nat Genet* **2001**; 29: 365-71.
- [30] Ball CA, Sherlock G, Parkinson H, et al. Standards for microarray data Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. *Science* **2002**; 298: 539.
- [31] Kuo WP, Jenssen TK, Butte AJ, Ohno-Machado L, Kohane IS. Analysis of matched mRNA measurements from two different microarray technologies. *Bioinformatics* **2002**; 18: 405-12.
- [32] Nimgaonkar A, Sanoudou D, Butte AJ, et al. Reproducibility of gene expression across generations of Affymetrix microarrays. *BMC Bioinformatics* **2003**; 4: 27.
- [33] Churchill GA. Fundamentals of experimental design for cDNA microarrays. *Nat Genet* **2002**; 32: 490-95.
- [34] Butte A. The use and analysis of microarray data. *Nat Rev Drug Discov* **2002**; 1: 951-60.
- [35] Schulze A, Downward J. Navigating gene expression using microarrays--a technology review. *Nat Cell Biol* **2001**; 3: E190-95.
- [36] Han ES, Wu Y, McCarter R, Nelson JF, Richardson A, Hilsenbeck SG. Reproducibility, sources of variability, pooling, and sample size: important considerations for the design of high-density oligonucleotide array experiments. *J Gerontol A Biol Sci Med Sci* **2004**; 59: 306-15.
- [37] Nelson PR, Goulter AB, Davis RJ. Effective analysis of genomic data. *Methods Mol Med* **2005**; 104: 285-312.
- [38] Dobbin K, Simon R. Sample size determination in microarray experiments for class comparison and prognostic classification. *Biostatistics* **2005**; 6: 27-38.
- [39] Jain A, Duin P, Mao J. Statistical pattern recognition: A review. *IEEE Transactions on PAMI* **2000**; 22: 4-37.
- [40] Troyanskaya O, Cantor M, Sherlock G, et al. Missing value estimation methods for DNA microarrays. *Bioinformatics* **2001**; 17: 520-25.
- [41] Ouyang M, Welsh WJ, Georgopoulos P. Gaussian mixture clustering and imputation of microarray data. *Bioinformatics* **2004**; 20: 917-23.
- [42] Zhou X, Wang X, Dougherty ER. Missing-value estimation using linear and non-linear regression with Bayesian gene selection. *Bioinformatics* **2003**; 19: 2302-07.
- [43] Wilson DL, Buckley MJ, Helliwell CA, Wilson IW. New normalization methods for cDNA microarray data. *Bioinformatics* **2003**; 19: 1325-32.
- [44] Yang YH, Dudoit S, Luu P, et al. Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Res* **2002**; 30: e15.
- [45] Zhao Y, Li MC, Simon R. An adaptive method for cDNA microarray normalization. *BMC Bioinformatics* **2005**; 6: 28.
- [46] Yoon D, Yi SG, Kim JH, Park T. Two-stage normalization using background intensities in cDNA microarray data. *BMC Bioinformatics* **2004**; 5: 97.
- [47] Fan J, Tam P, Woude GV, Ren Y. Normalization and analysis of cDNA microarrays using within-array replications applied to neuroblastoma cell response to a cytokine. *Proc Natl Acad Sci U S A* **2004**; 101: 1135-40.

- [48] Futschik M, Crompton T. Model selection and efficiency testing for normalization of cDNA microarray data. *Genome Biol* **2004**; 5: R60.
- [49] Park T, Yi SG, Kang SH, Lee S, Lee YS, Simon R. Evaluation of normalization methods for microarray data. *BMC Bioinformatics* **2003**; 4: 33.
- [50] Smyth GK, Speed T. Normalization of cDNA microarray data. *Methods* **2003**; 31: 265-73.
- [51] Freudenberg J, Boriss H, Hasenclever D. Comparison of preprocessing procedures for oligo-nucleotide micro-arrays by parametric bootstrap simulation of spike-in experiments. *Methods Inf Med* **2004**; 43: 434-38.
- [52] Irizarry RA, Hobbs B, Collin F, *et al*. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics* **2003**; 4: 249-64.
- [53] Irizarry RA, Bolstad BM, Collin F, Cope LM, Hobbs B, Speed TP. Summaries of Affymetrix GeneChip probe level data. *Nucl Acids Res* **2003**; 31: e15.
- [54] Bolstad BM, Irizarry RA, Astrand M, Speed TP. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* **2003**; 19: 185-93.
- [55] Ding Y, Wilkins D. The effect of normalization on microarray data analysis. *DNA Cell Biol* **2004**; 23: 635-42.
- [56] Hoffmann R, Seidl T, Dugas M. Profound effect of normalization on detection of differentially expressed genes in oligonucleotide microarray data analysis. *Genome Biol* **2002**; 3: Research33.
- [57] Parrish RS, Spencer HJ. 3rd. Effect of normalization on significance testing for oligonucleotide microarrays. *J Biopharm Stat* **2004**; 14: 575-89.
- [58] Herrero J, Diaz-Urriarte R, Dopazo J. Gene expression data preprocessing. *Bioinformatics* **2003**; 19: 655-6.
- [59] Durbin BP, Hardin JS, Hawkins DM, Rocke DM. A variance-stabilizing transformation for gene-expression microarray data. *Bioinformatics* **2002**; 18: S105-10.
- [60] Ekstrom CT, Bak S, Kristensen C, Rudemo M. Spot shape modelling and data transformations for microarrays. *Bioinformatics* **2004**; 20: 2270-78.
- [61] Huber W, von Heydebreck A, Sultmann H, Poustka A, Vingron M. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics* **2002**; 18: S96-104.
- [62] Geller SC, Gregg JP, Hagerman P, Rocke DM. Transformation and normalization of oligonucleotide microarray data. *Bioinformatics* **2003**; 19: 1817-23.
- [63] Thygesen HH, Zwinderman AH. Comparing transformation methods for DNA microarray data. *BMC Bioinformatics* **2004**; 5: 77.
- [64] Duda R, Hart P, Stork D. *Pattern Classification*, 2nd Edition ed, Wiley, NY 2001.
- [65] Kulkarni SR, Lugosi G, Venkatesh SS. Learning pattern classification - a survey. *IEEE Trans Inform Theory* **1998**; 44: 2178-206.
- [66] Statnikov A, Aliferis CF, Tsamardinos I, Hardin D, Levy S. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics* **2005**; 21: 631-43.
- [67] Khan J, Wei JS, Ringner M, *et al*. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nat Med* **2001**; 7: 673-79.
- [68] Breiman L, Friedman JH, Olshen R, Stone CJ. *Classification and Regression Trees*, Wadsworth, Belmont, CA 1984.
- [69] Vapnik V. *Statistical Learning Theory*, Chichester, Wiley, UK 1998.
- [70] Jain A, Zongker D. Feature Selection: Evaluation, Application, and Small Sample Performance. *IEEE Trans Pattern Anal Mach Intell* **1997**; 19: 153-58.
- [71] Friedman JH. Regularized Discriminant Analysis. *J Am Statistical Assoc* **1989**; 84: 165-75.
- [72] Chen Y, Dougherty E, Bittner M. Ratio-based decisions and the quantitative analysis of cDNA microarray images. *J Biomedical Optics* **1997**; 2: 364-67.
- [73] Su Y, Murali TM, Pavlovic V, Schaffer M, Kasif S. RankGene: identification of diagnostic genes based on expression data. *Bioinformatics* **2003**; 19: 1578-79.
- [74] Wang Y, Makedon FS, Ford JC, Pearlman J. HykGene: a hybrid approach for selecting marker genes for phenotype classification using microarray gene expression data. *Bioinformatics* **2005**; 21: 1530-37.
- [75] Dudoit S, Yang YH, Callow MJ, Speed TP. Statistical Methods for Identifying Differentially Expressed Genes In Replicated cDNA Microarray Experiments. *Statistica Sinica* **2002**; 12: 111-39.
- [76] Tsai CA, Chen CH, Lee TC, Ho IC, Yang UC, Chen JJ. Gene selection for sample classifications in microarray experiments. *DNA Cell Biol* **2004**; 23: 607-14.
- [77] Pan W. A comparative review of statistical methods for discovering differentially expressed genes in replicated microarray experiments. *Bioinformatics* **2002**; 18: 546-54.
- [78] Troyanskaya OG, Garber ME, Brown PO, Botstein D, Altman RB. Nonparametric methods for identifying differentially expressed genes in microarray data. *Bioinformatics* **2002**; 18: 1454-61.
- [79] Tsai CA, Hsueh HM, Chen JJ. Estimation of false discovery rates in multiple testing: application to gene microarray data. *Biometrics* **2003**; 59: 1071-81.
- [80] Simon R. Diagnostic and prognostic prediction using gene expression profiles in high-dimensional microarray data. *Br J Cancer* **2003**; 89: 1599-604.
- [81] Pan W. On the use of permutation in and the performance of a class of nonparametric methods to detect differential gene expression. *Bioinformatics* **2003**; 19: 1333-40.
- [82] Raudys SJ, Jain AK. Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners. *IEEE Trans Pattern Anal Mach Intell* **1991**; 13: 252-64.
- [83] Levner I. Feature selection and nearest centroid classification for protein mass spectrometry. *BMC Bioinformatics* **2005**; 6: 68.
- [84] Bo T, Jonassen I. New feature subset selection procedures for classification of expression profiles. *Genome Biol* **2002**; 3: 1-11.
- [85] Stone M. Cross-validatory choice and assessment of statistical predictions. *J Royal Stat Soc Ser B* **1974**; 36: 111-47.
- [86] Lachenbruch P, Mickey A. Estimation of error rates in discriminant analysis. *Technometrics* **1968**; 10: 1-11.
- [87] Lunts AL, Brailovsky VL. Evaluation of attributes obtained in statistical decision rules. *Eng Cybernetics* **1967**; 3: 98-109.
- [88] Braga-Neto UM, Dougherty ER. Is cross-validation valid for small-sample microarray classification? *Bioinformatics* **2004**; 20: 374-80.
- [89] Evgeniou T, Pontil M, Elisseeff A. Leave-one-out error, stability, and generalization of voting combination of classifiers. *Mach Learn* **2004**; 55: 71-97.
- [90] Davison A, Hall P. On the bias and variability of bootstrap and cross-validation estimates of error rate in discriminant problems. *Biometrika* **1992**; 79: 279-84.
- [91] Jain AK, Dubes RC, Chen CC. Bootstrap techniques for error estimation. *IEEE Trans Pattern Anal Mach Intell* **1987**; 9: 628-33.
- [92] Efron BaT, R.J. *An Introduction to the Bootstrap*, Chapman & Hall, London 1993.
- [93] Ambroise C, McLachlan GJ. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc Natl Acad Sci USA* **2002**; 99: 6562-66.
- [94] Lukashin AV, Fuchs R. Analysis of temporal gene expression profiles: clustering by simulated annealing and determining the optimal number of clusters. *Bioinformatics* **2001**; 17: 405-14.
- [95] Ghosh D, Chinnaiyan AM. Mixture modelling of gene expression data from microarray experiments. *Bioinformatics* **2002**; 18: 275-86.
- [96] Selim SZ, Ismail MA. K-Means-Type algorithms: a generalized convergence theorem and characterization of local optimality. *IEEE Trans Pattern Anal Mach Intell PAMI* **1984**; 6: 81-87.
- [97] Martinez WL, Martinez AR. *Computational Statistics Handbook with MATLAB*, CRC Press, Boca Raton, 2001.
- [98] Asyali MH, Alci M. Reliability analysis of microarray data using fuzzy C-means and normal mixture modeling based classification methods. *Bioinformatics* **2005**; 21: 644-49.
- [99] Bezdek JC. *Pattern Recognition with Fuzzy Objective Function Algorithm*, Plenum Press, New York 1981.
- [100] Dembele DaK, P. Fuzzy c-means method for clustering microarray data. *Bioinformatics* **2003**; 19: 973-80.
- [101] Yeung KY, Fraley C, Murua A, Raftery AE, Ruzzo WL. Model-based clustering and data transformations for gene expression data. *Bioinformatics* **2001**; 17: 977-87.
- [102] McLachlan GJ, Bean RW, Peel D. A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics* **2002**; 18: 413-22.

- [103] McLachlan GJ. The classification and mixture maximum likelihood approaches to cluster analysis, In: Krishnaiah PR, Kanal LN, Eds, Handbook of Statistics. Amsterdam, North-Holland 1982; 199-208.
- [104] McLachlan GJ, Basford KE. Mixture Models, Inference and Applications to Clustering. Marcel Dekker, New York 1989.
- [105] Symons M. Clustering criteria and multivariate normal mixtures. *Biometrics* **1981**; 37: 35-43.
- [106] Wolfe J. Pattern Clustering by Multivariate Mixture Analysis. *Multivar Behav Res* **1970**; 5: 329-50.
- [107] Dempster A, Laird N, Rubin D. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J Royal Stat Soc* **1977**; B39: 1-38.
- [108] Redner R, Walker H. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Rev* **1984**; 26: 195-202.
- [109] Moon TK. The Expectation-Maximization Algorithm. *IEEE Sign Proc Mag* **1996**; 13: 47-60.
- [110] McLachlan GJ, Basford KE. Mixture Models, Marcel Dekker, New York 1988.
- [111] Eisen MB, Spellman PT, Brown PO, Botstein D. Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci USA* **1998**; 95: 14863-68.
- [112] Tavazoie S, Hughes JD, Campbell MJ, Cho RJ, Church GM. Systematic determination of genetic network architecture. *Nat Genet* **1999**; 22: 281-85.
- [113] McLachlan GJ, Bean RW, Peel D. A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics* **2002**; 18: 413-22.
- [114] Schwarz G. Estimating the dimension of a model. *Ann Stat* **1978**; 6: 461-64.
- [115] Sharan R, Maron-Katz A, Shamir R. CLICK and EXPANDER: a system for clustering and visualizing gene expression data. *Bioinformatics* **2003**; 19: 1787-99.
- [116] Ben-Dor A, Shamir R, Yakhini Z. Clustering gene expression patterns. *J Comput Biol* **1999**; 6: 281-97.
- [117] Jiang D, Pei J, Zhang A. DHC: A Density-Based Hierarchical Clustering Method for Time-Series Gene Expression Data. *Proc BIBE 2003: Third IEEE Intl Symp Bioinformatics and Bioeng* **2003**; 393-400.
- [118] Xu R, Wunsch D, II. Survey of Clustering Algorithms. *IEEE Trans on Neural Networks* **2005**; 16: 645-78.
- [119] Jiang D, Tang C, Zhang A. Cluster analysis for gene expression data: a survey. *Knowl Data Eng IEEE Trans on* **2004**; 16: 1370-86.
- [120] Herwig R, Poustka AJ, Muller C, Bull C, Lehrach H, O'Brien J. Large-scale clustering of cDNA-fingerprinting data. *Genome Res* **1999**; 9: 1093-105.
- [121] Tamayo P, Slonim D, Mesirov J, et al. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc Natl Acad Sci USA* **1999**; 96: 2907-12.
- [122] Wu LF, Hughes TR, Davierwala AP, Robinson MD, Stoughton R, Altschuler SJ. Large-scale prediction of *Saccharomyces cerevisiae* gene function using overlapping transcriptional clusters. *Nat Genet* **2002**; 31: 255-65.
- [123] Cheng Y, Church GM. Biclustering of expression data. *Proc Int Conf Intell Syst Mol Biol* **2000**; 8: 93-103.
- [124] Hartigan JA. Direct Clustering of a Data Matrix. *J Am Statistical Assoc (JASA)* **1972**; 67: 123-29.
- [125] Madeira SC, Oliveira AL. Biclustering algorithms for biological data analysis: a survey. *Comput Biol Bioinformatics IEEE/ACM Trans* **2004**; 1: 24-45.
- [126] Lazzeroni L, Owen A. Plaid Models for Gene Expression Data. *Statistica Sinica* **2002**; 12: 61-86.
- [127] Tanay A, Sharan R, Shamir R. Discovering statistically significant biclusters in gene expression data. *Bioinformatics* **2002**; 18: S136-44.
- [128] Yang J, Wang W, Wang H, Yu P. Delta-clusters: Capturing subspace correlation in a large data set. In: *Proceedings of the 18th International Conference on Data Engineering*, **2002**; 517-28.
- [129] Getz G, Levine E, Domany E. Coupled two-way clustering analysis of gene microarray data. *Proc Natl Acad Sci USA* **2000**; 97: 12079-84.
- [130] Getz G, Gal H, Kela I, Notterman DA, Domany E. Coupled two-way clustering analysis of breast cancer and colon cancer gene expression data. *Bioinformatics* **2003**; 19: 1079-89.
- [131] Sheng Q, Moreau Y, De Moor B. Biclustering microarray data by Gibbs sampling. *Bioinformatics* **2003**; 19: II196-II205.
- [132] Yang J, Wang H, Wang W, Yu P. Enhanced biclustering on expression data. In: *Bioinformatics and Bioengineering, 2003 Proceedings Third IEEE Symposium on*, **2003**; 321-27.
- [133] Yang J, Wang W, Wang H, Yu P. Delta-clusters: capturing subspace correlation in a large data set. In: *Data Engineering, 2002 Proceedings 18th International Conference on*, **2002**; 517-28.
- [134] Getz G, Domany E. Coupled two-way clustering server. *Bioinformatics* **2003**; 19: 1153-54.
- [135] Blat M, Wiseman S, Domany E. Super-Paramagnetic Clustering of Data. *Phys Rev Lett* **1996**; 76: 3251-55.
- [136] Tanay A, Sharan R, Kupiec M, Shamir R. Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proc Natl Acad Sci USA* **2004**; 101: 2981-6. Epub 004 Feb 18.
- [137] Wu CJ, Fu Y, Murali TM, Kasif S. Gene expression module discovery using Gibbs sampling. *Genome Inform Ser Workshop Genome Inform* **2004**; 15: 239-48.
- [138] Liu J, Wang W, Yang J. Gene ontology friendly biclustering of expression profiles. In: *Comput Syst Bioinformatics Conference, 2004 CSB 2004 Proceedings 2004 IEEE*, **2004**; 436-47.
- [139] Herrero J, Vaquerizas JM, Al-Shahrour F, et al. New challenges in gene expression data analysis and the extended GEPAS. *Nucl Acids Res* **2004**; 32: W485-91.
- [140] Choi JK, Choi JY, Kim DG, et al. Integrative analysis of multiple gene expression profiles applied to liver cancer study. *FEBS Lett* **2004**; 565: 93-100.
- [141] Feng W, Wang G, Zeeberg BR, et al. Development of gene ontology tool for biological interpretation of genomic and proteomic data. *AMIA Annu Symp Proc* **2003**; 4: 839.
- [142] Zeeberg BR, Feng W, Wang G, et al. GoMiner: a resource for biological interpretation of genomic and proteomic data. *Genome Biol* **2003**; 4: R28.
- [143] Guo Z, Zhang T, Li X, et al. Towards precise classification of cancers based on robust gene functional expression profiles. *BMC Bioinformatics* **2005**; 6: 58.
- [144] Cheng J, Cline M, Martin J, et al. A Knowledge-Based Clustering Algorithm Driven by Gene Ontology. *J Biopharm Stat* **2004**; 14: 687-700.
- [145] Al-Shahrour F, Diaz-Uriarte R, Dopazo J. Discovering molecular functions significantly related to phenotypes by combining gene expression data and biological information. *Bioinformatics* **2005**; 19: 2988-93.