33. Statistical Methodologies for Analyzing Genomic Data

The purpose of this chapter is to describe and review a variety of statistical issues and methods related to the analysis of microarray data. In the first section, after a brief introduction of the DNA microarray technology in biochemical and genetic research, we provide an overview of four levels of statistical analyses. The subsequent sections present the methods and algorithms in detail.

In the second section, we describe the methods for identifying significantly differentially expressed genes in different groups. The methods include fold change, different *t*-statistics, empirical Bayesian approach and significance analysis of microarrays (SAM). We further illustrate SAM using a publicly available colon-cancer dataset as an example. We also discuss multiple comparison issues and the use of false discovery rate.

In the third section, we present various algorithms and approaches for studying the relationship among genes, particularly clustering and classification. In clustering analysis, we discuss hierarchical clustering, *k*-means and probabilistic model-based clustering in detail with examples. We also describe the adjusted Rand index as a measure of agreement between different clustering methods. In classification analysis, we first define some basic concepts related to classification. Then we describe four commonly used classification methods including linear discriminant analysis (LDA), support vector machines (SVM), neural network and tree-and-forest-based classification. Examples are included to illustrate SVM and tree-and-forest-based classification.

The fourth section is a brief description of the meta-analysis of microarray data in three different settings: meta-analysis of the same biomolecule and same platform microarray data, meta-analysis of the same biomolecule but different platform microarray data, and meta-analysis of different biomolecule microarray data.

We end this chapter with final remarks on future prospects of microarray data analysis.

Since the seminal work on microarray technology of *Schena* et al. [33.1], microarray data have attracted a great deal of attention, as reflected by the ever increasing number of publications on this technology in the past decade. The applications of the microarray technology encompass many fields of science from the search for differentially expressed genes [33.2], to the understanding of regulatory networks [33.3], DNA sequencing and mutation study [33.4], single nucleotide polymorphism (SNP) detection [33.5], cancer diagnosis [33.6], and drug discovery [33.7].

Accompanying the advancement of the microarray technology, analyzing microarray data has arguably become the most active research area of statistics and bioinformatics. Figure 33.1 provides a four-level overview of the analytic process. The first challenge in dealing with the microarray data is to preprocess the data, which involves background subtraction, array normalization, and probe-level data summarization. The purpose of this preprocessing is to remove noise and artifacts in order to enhance and extract hybridization signals. This data preprocessing is also often referred as

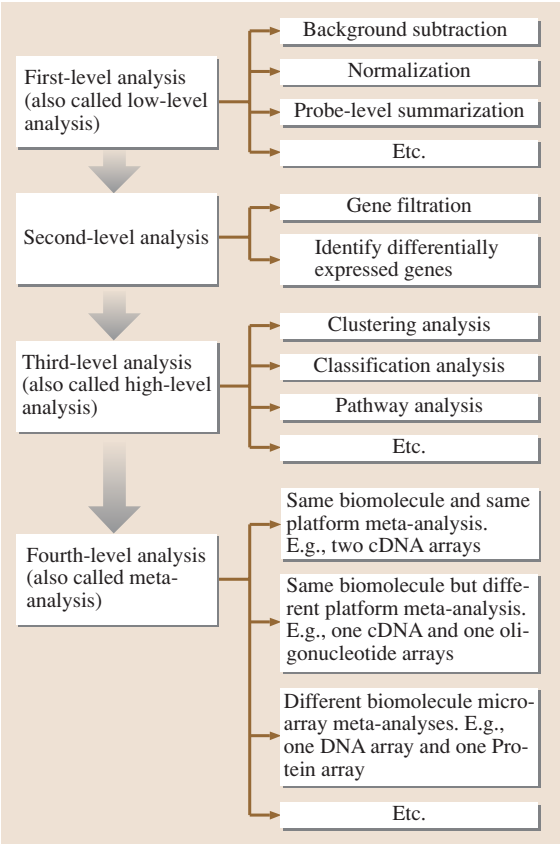

Fig. 33.1 Diagram of the four-level analysis of microarray data

the low-level analysis [33.8]. After the data are processed and cleaned, they are analyzed for different purposes. The focus of this article is on the methods for this postprocessing analysis.

The second-level analysis usually contains two steps: one is to filter *unusual* genes whose expression profiles are suspicious due to noise or are too extreme, and the other is to identify the differentially expressed genes across different samples. The gene filtration process is generally heuristic and specific to known biological contents. Thus, we will not discuss it here. To identify genes that have significantly different expression profiles, the commonly used approaches include the estimation of fold change, Student's T-test, the Wilcoxon rank sum test, the penalized T-test, empirical Bayes [33.9], and significance analysis of microarray (SAM, *Tusher* et al. [33.10]). We will review these methods in Sect. 33.1.

We will review the third-level analysis in Sect. 33.2. This type of analysis is also called high-level analysis [33.11], and it includes clustering, classification and pathway analysis. This is usually conducted on a subset of genes that are selected from the second-level analysis. To identify genes that may be correlated to each other, clustering analysis has become particularly popular, and the approaches include hierarchical clustering [33.12], k-means [33.13], self-organization maps (SOM) [33.14], principle-component analysis (PCA) [33.15], and probabilistic model-based clustering [33.16].

To classify tissue samples or diagnose diseases based on gene expression profiles, both classic discriminant analysis and contemporary classification methods have been used and developed. The methods include k-nearest neighbors (KNN) [33.17], linear discriminant analysis (LDA) [33.18], support vector machine (SVM) [33.19], artificial neural networks (ANN) [33.20], classification trees [33.21], and random and deterministic forests [33.18]. It is noteworthy that tree- and forest-based approaches can be easily applied to the entire microarray dataset without restricting our attention to a subset of selected genes.

To identify genes that may be on the same pathway of a particular biological process, relevance networks [33.22], linear differential equation [33.23], Boolean networks [33.24], Bayesian networks [33.25] and the probabilistic rational model (PRM) [33.26] have been used and developed.

The fourth-level analysis, also referred as meta-analysis, is a relative new topic for the analysis of microarray data. Because many different types and platforms of microarrays can be designed to address the same (or similar) biological problems, it is useful to compare and synthesize the results from different studies.

Before we introduce specific methods, we should point out that, as a result of high-throughput technology, the unique challenge from analyzing microarray data is the large number of genes (tens of thousands) and relatively small sample sizes (commonly on the order of tens or hundreds). In this article, n denotes the number of genes and m the number of arrays. n is generally much greater than m.

33.1 Second-Level Analysis of Microarray Data

33.1.1 Notation

For a two-channel cDNA microarray data [33.1], we have a $2n \times m$ matrix of imaging data reflecting the red (cy5) and green (cy3) signals for each of the n genes on m arrays. The log ratio of the red to green signal is usually taken for each gene, and the analysis will be based on an $n \times m$ data matrix.

For one-channel Affymetrix Oligonucleotide GeneChip data [33.27], we have a $2\sum_{i=1}^{n} p_i \times m$ matrix of raw image data where p_i is the number of probes for the i-th gene. Note that, for each probeset, Affymetrix uses a pair of perfect match (PM) and mismatch (MM). As for oligonucleotide microarrays, steps [differences, ratios, analysis of variance (ANOVA) models, etc.] can be taken to summarize the PM and MM signals for each gene, and we still have an $n \times m$ data matrix.

A major objective of microarray analysis is to infer significantly differentially expressed genes (abbreviated as SDE genes) across different samples, e.g., m_1 tumor samples versus m_2 normal samples.

Let $Y_{ij,k}$ be the expression level of the i-th gene on the j-th array in the k-th sample. Let $\overline{Y_{i.,1}}$ and $\overline{Y_{i.,2}}$ denote the average expression level of the i-th gene in samples 1 and 2, respectively.

33.1.2 Fold Change

Many studies identify SDE genes in two samples based on simple fold-change thresholds such as a two-fold change in means. Although the choice of a threshold is somewhat arbitrary, fold change is intuitive and biologically meaningful, and serves as an effective preliminary step to eliminate a large portion of genes whose data are of little interest in a particular study.

33.1.3 *t*-Statistic

As in many clinical studies, the t-statistic provides a simple, extremely useful tool to compare the data from two samples. Let $\overline{M}$ be the mean difference between the expression profiles of a gene in two groups and $\mathrm{se}(\overline{M})$ be the standard error of $\overline{M}$. The t-statistic, defined as

$$t = \frac{\overline{M}}{\mathrm{sd}(\overline{M})} ,$$

is useful to test a null hypothesis that the gene is not differentially expressed in the two groups against the alternative hypothesis that the gene is differentially expressed.

Unlike a typical clinical study, in which we have one pair or a very few pairs of hypotheses to test, in microarray analysis we have a pair of hypotheses for every gene of interest. This means that we inevitably deal with the multiple comparison issue. Although this issue is difficult and there is no clear-cut, ideal answer, many reasonable solutions have been proposed.

Efron et al. [33.9] proposed to inflate $\mathrm{se}(\overline{M})$ by adding a constant that equals the 90-th percentile of the standard errors of all the genes. *Tusher* et al. [33.10] call such a constant a fudge factor, and propose to estimate it by minimizing the coefficient of variation of the absolute t-values. We will discuss this approach in detail in Sect. 33.1.4. Other approaches have also been proposed; for example, *Smyth* [33.28] replaces $\mathrm{se}(\overline{M})$ with a Bayesian shrinkage estimator of the standard deviation.

The permutation test is also commonly used to compare the microarrays. Permutations are usually performed at the array level to create a situation similar to the null hypothesis while maintaining the dependence structure among the genes [33.10]. In every permutation, a t-statistic can be calculated for each gene. Once a large number of permutations are completed, we have an empirical distribution for the t-statistic under the null hypothesis, which then can be used to identify SDE genes.

33.1.4 The Multiple Comparison Issue

As we mentioned earlier, we have to control the type I error rate α while testing a large number of hypotheses simultaneously. There are two commonly used approaches to deal with this issue. One is to control the family-wise error rate (FWER) and the other is to control the false discovery rate (FDR).

The FWER controls the probability of making at least one false positive call at the desired significance level. FWER guarantees that the type I error rate is less than or equal to a specified value for any given set of genes. The most known example of FWER is Bonferroni correction that divides the desired significance level α by total number of hypotheses. If the desired significance level is 0.05 and we compare expression profiles in 10 000 genes, a gene is declared to have significantly different profiles in two groups if the P-value is not greater than $\frac{0.05}{10\,000} = 5 \times 10^{-6}$. Another FWER approach is the so-called *Šidák* correction in which the

adjusted type I error rate is at $1-(1-\alpha)^{\frac{1}{n}}$ [33.29], which is close to α/n. Clearly, Bonferroni and Šidák corrections are sufficient but not necessary conditions [33.30], and FWER approaches are generally very conservative and set a stringent bar to declare SDE genes.

Because of the conservative nature of the FWER approaches, the FDR concept has flourished since it was proposed by [33.31]). FDR is defined as the mean of the ratio of the number, denoted by V, of falsely rejected hypotheses to the total rejected hypotheses, denoted by R, namely,

$$\mathrm{FDR} = E\left(\frac{V}{R}|R>0\right)\Pr(R>0)\,,$$

where $\Pr(R>0)$ is the probability of rejecting at least one hypothesis.

The FDR can be controlled at a given α level through the following steps. First, for n genes, we have n null hypotheses and n p values, denoted by $p_1,\ldots,p_n$. Then, we sort the p-values in ascending order such that $p_{(1)}\le\cdots\le p_{(n)}$. We reject any gene i that satisfies the condition $p_{(i)}\le\frac{i}{n}\times\frac{\alpha}{p_0}$, where p_0 is the proportion of genes for which the null hypotheses are indeed true. Because p_0 is unknown in practice, the most conservative approach is to replace it with 1. Recently, attempts have been made to estimate p_0 as in Tusher et al.'s SAM, where they used a permutation procedure to estimate p_0. Similar to the classical p-values, the significance measures for each gene in terms of FDR are called q-values, a name that was introduced by *Storey* [33.32, 33].

In addition, the FDR concept has been generalized. For example, *Storey* and *Tibshirani* [33.9] and *Storey* et al. [33.32] proposed positive FDR (pFDR), which corrects the error rate only when they are positive findings. For microarray data, many gene profiles are correlated, *Troendle* [33.34] proposed an adjusted FDR to address the correlation and demonstrated the benefit in terms of gained power.

33.1.5 Empirical Bayesian Approach

Using microarray data from a breast cancer study, *Efron* et al. [33.9, 35] described the empirical Bayesian method. As an initial step, a summary statistic, Z, needs to be defined for every gene to reflect the scientific interest; this can be the t-statistic as described above, a Wilcoxon rank statistic, or another choice. All genes are perceived to belong to either the differentially or nondifferentially expressed group. The density of Z_i is $f_0(z_i)$ if gene i is in the nondifferentially expressed group, and $f_1(z_i)$ otherwise. Without knowing the group, Z_i has the following mixture distribution:

$$p_0 f_0(z_i)+p_1 f_1(z_i)\,,$$

where p_0 is the prior probability that gene i is not differentially expressed, and $p_1=1-p_0$.

Based on Bayes' theorem, the posterior probability that gene i is not differentially expressed given Z_i is

$$p_0(z_i)=p_0\frac{f_0(z_i)}{f(z_i)}\,.$$

We can estimate the mixture density $f(z_i)$ by the empirical distribution $\hat{f}(z_i)$ because the genes of interest are naturally a mixture of the two groups. In addition, the null density $f_0(z_i)$ can be estimated through the permutation that artificially generates data under the null hypothesis. In other words, we can derive the posterior probability $p_0(z_i)$ for a given prior p_0. The choice of p_0 can be subjective. One conservative possibility is to choose p_0 to be the minimum of $\hat{f}(z_i)/\hat{f}_0(z_i)$ so that the posterior probability $p_1(z_i)$ that gene i is differentially expressed is non-negative. Note that $p_1(z_i)=1-p_0(z_i)$. Finally, all genes can be ranked according to $p_1(z_i)$ and highly probably differentially expressed genes can be selected.

Efron et al. [33.9, 35] did not assume a specific form for $f(z_i)$. In contrast, *Lonnstedt* and *Speed* [33.36] assumed that the data comes from the mixture of normal distributions and used the conjugate priors for the variances and the means. Under those assumptions, they derived the log odds posterior test. *Smyth* [33.28] extended the hierarchical model of *Lonnstedt* and *Speed* [33.36] to deal with microarray experiments with more than two sample groups. The method is called the Limma algorithm.

33.1.6 Significance Analysis of Microarray (SAM)

Tusher et al. [33.10] introduced the SAM algorithm. SAM identifies genes with statistically significant changes in expression by assimilating a set of gene-specific t-tests in which the standard error is adjusted by adding a small positive constant. It performs a random permutation among experiments and declares the significant genes based on a selected threshold. For the given threshold, SAM estimates the FDR by comparing the number of genes significant in the permuted samples with the number of genes significant in the original sample.

SAM can be downloaded from http://www-stat.stanford.edu/~tibs/SAM/. Specifically, first, for each gene i, SAM computes a t-like statistic

$$t_i = \frac{r_i}{s_i + s_0} ,$$

where r_i is the difference between the expression means of gene i in the two groups (expression is on a logarithm scale), s_i is the standard error, and s_0 is the fudge factor to be estimated. Secondly, similarly to the FDR scheme, all t_i values are sorted into the order statistics

$$t_{(1)} \le t_{(2)} \le \ldots \le t_{(n)} .$$

To choose the significance threshold, the expression data are permuted in the two groups within each gene B times, and during each permutation, we repeat the first two steps, which leads to a set of order statistics:

$$t^b_{(1)} \le t^b_{(2)} \le \ldots \le t^b_{(n)} .$$

After the permutations, we calculate the mean of the order statistics for each gene as follows

$$\bar{t}_{(i)} = \frac{1}{B} \sum_{b=1}^{B} t^b_{(i)} .$$

For a given threshold Δ, a gene is considered significant if $|t_{(i)} - \bar{t}_{(i)}| > \Delta$, and the FDR is estimated by the ratio of the number of genes found to be significant in the permutation samples to the number of genes called significant in the original sample.

Example 1: Identification of SDE Genes Using SAM. In this example, we apply SAM to examine a publicly available colon-cancer dataset [33.37]. This dataset contains the expression profiles of 2000 genes using an Affymetrix oligonucleotide array in 22 normal and 40 colon-cancer tissues.

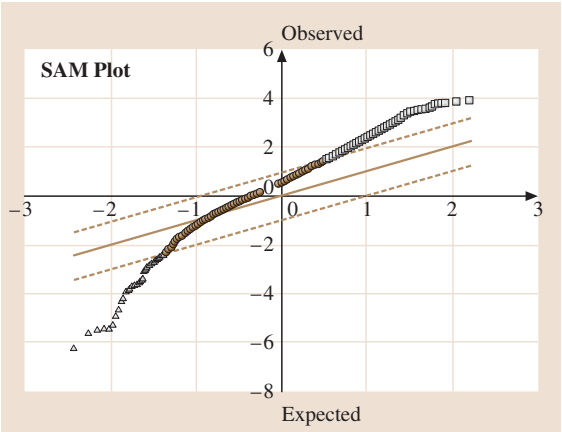

Fig. 33.2 The quantile–quantile plot from SAM for the colon-cancer dataset. Genes are declared significantly changed when their corresponding t-values are outside the two dashed lines. The white square and triangle points correspond to the genes that are significantly overexpressed and underexpressed, respectively

Figure 33.2 displays the quantile–quantile plot from SAM. The two dashed lines determine a boundary to call genes SDE depending on the choice of Δ. For example, Δ was chosen as 0.9857 in Fig. 33.2 to control the FDR at about 5%. The white square and triangle points in the figure correspond to the genes that are declared to be significantly overexpressed and underexpressed respectively. Out of the 490 declared SDE genes (440 overexpressed and 50 underexpressed), 25 genes are expected to be declared falsely.

33.2 Third-Level Analysis of Microarray Data

The third-level microarray analysis includes clustering, classification and pathway analysis. These approaches usually, though not always, follow the second-level microarray analysis because most of them can work effectively on only a small number of genes.

33.2.1 Clustering

Clustering is arguably the most commonly used approach at the third-level of analysis [33.38, 39]. It is an unsupervised learning algorithm from a machine-learning viewpoint, because the gene classes are unknown or not used, and need to be *discovered* from the data. Therefore, the goal of clustering analysis is to group genes (or arrays) based on their similarity in the feature space (e.g., expression pattern).

The underlying assumption behind clustering is that genes with similar expression profiles should share some common biological behaviors, e.g., belonging to the same protein complex or gene family [33.40], having common biological functions [33.41], being regulated by common transcription factors [33.3], belonging to the same genetic pathway, or coming from the same origin [33.39].

After the clusters are formed, a dendrogram or a tree of all genes will be viewed, although the views are not unique, because there is a left-or-right selection at each splitting step. Two popular programs for gene clustering are *Eisen* et al.'s TreeView program [33.12] and Li and Wong's dChip programs [33.8]. Routines are also available in standard statistical packages such as R, Splus, and SAS.

Distance

In order to group objects (genes or arrays) together, we need to define a measure to quantify the similarity among objects in the feature space. Such a measure of similarity is called a distance. There are several commonly used definitions of distance. Suppose that the expression profiles of two genes are $Y_i = (y_{i1}, y_{i2}, \ldots, y_{im})$ and $Y_j = (y_{j1}, y_{j2}, \ldots, y_{jm})$.

The Euclidean distance between Y_i and Y_j is

$$d_{\mathrm{E}}\left(Y_i, Y_j\right) = \left[\sum_{k=1}^{m}\left(y_{ik} - y_{jk}\right)^2\right]^{\frac{1}{2}} .$$

The city-block distance between Y_i and Y_j is

$$d_{\mathrm{C}}\left(Y_i, Y_j\right) = \sum_{k=1}^{m}\left|y_{i1} - y_{j1}\right| .$$

The Pearson correlation distance between Y_i and Y_j is

$$d_{\mathrm{R}}\left(Y_i, Y_j\right) = 1 - r_{Y_i Y_j} ,$$

where $r_{Y_i Y_j}$ is the Pearson correlation coefficient between Y_i and Y_j.

The Spearman correlation distance between Y_i and Y_j uses the rank-based correlation coefficient in which the expression levels are replaced with the ranks.

More definitions can be found in the book by *Draghici* [33.30]. We should note that the Euclidean and city-block distance look for similar expression numerical values while the Pearson and Spearman distances tend to emphasize similar expression patterns.

The distances defined above measure the gene-wise distance. When clusters are found, we also need to define the distance between two clusters. The four approaches are: single linkage distance (the minimum distance between any gene in one cluster and any gene in the other cluster), complete linkage distance (the maximum distance between any gene in one cluster and any gene in the other cluster), average linkage distance (the average of all pair-wise distances between any gene in one cluster and any gene in the other cluster), and centroid linkage distance (the distance between the centroids of the two clusters).

Clustering Methods

When a distance measure is chosen, there are different ways to execute the clustering process. The clustering methods broadly fall into two categories: hierarchical methods and partitioning methods. Hierarchical methods build up a hierarchy for clusters, from the lowest one (all genes are in one cluster) to the highest one (all genes are in their own clusters) while partitioning methods group the genes into the different clusters based on their expression profiles. Therefore, one does not need to provide the cluster number for hierarchical clustering methods but it is necessary for the partitioning clustering methods.

Hierarchical methods include agglomerative hierarchical methods and divisive hierarchical methods.

The agglomerative hierarchical methods use a bottom-up strategy by treating each individual gene as a cluster at the first step. Then two nearest genes are found and assigned into a cluster where the *nearest* is defined by the distance between these two genes, e.g., for a Pearson distance nearest means the two genes having the largest correlation coefficient. Then an agglomerative hierarchical method assigns a new expression profile for the formed clusters, and repeats these steps until there is only one cluster left.

The divisive hierarchical methods, on the other hand, treat all genes belonging to one cluster at the beginning. Then in each step they choose a partitioning method to divide all genes into a predecided number of clusters, e.g., using k-means to partition genes into two clusters at each single step. Therefore, the decisive hierarchical clustering methods employ the bottom-down strategy.

The k-means clustering is the simplest and fastest clustering algorithm [33.42] among the partitioning methods. It has been widely used in many microarray analyses. To form K clusters, the k-means algorithm allocates the observations into different groups in order to minimize the within-group sum of squares

$$\min_{S_K}\left[\sum_{k=1}^{K}\sum_{i\in S_K}\sum_{j=1}^{m}\left(y_{ij} - \overline{y}_{kj}\right)^2\right] ,$$

where K is the prespecified cluster number, S_k is the set of objects in the k-th cluster and $\overline{y}_{kj}$ is the mean of group j in cluster k. In other words, k-means clustering uses the Euclidean distance.

The k-means clusters are formed through iterations as follows: First, k center genes are randomly selected, and every other gene is assigned to the closest center gene. Then, the center is redefined for each cluster to

minimize the sum of squares toward the center. In fact, the coordinates of a cluster center are the mean expressions of all the genes in that cluster. After the centers are redefined, all genes are regrouped and the iteration process continues until it converges.

After analyzing a yeast cell-cycle expression dataset, *Duan* and *Zhang* [33.43] noted that it could be particularly useful to use a weighted sum of squares for gene clustering to take into account the loss of synchrony of cells. We refer to *Duan* and *Zhang* [33.43] for the details.

Another widely used partitioning clustering algorithm is *self-organizing maps* (SOMs) which were developed by *Kohonen* [33.44]. In essence, SOM clustering is a spatial version of the k-means clustering. For a prespecified grid (i. e., a 6×8 hexagonal grid), SOMs project high-dimensional gene expression data onto a two- or three-dimensional map and place similar genes close to each other. Here, the centroid positions of clusters are related to one another via a spatial topology (e.g., the squared map), and are also iteratively adjusted according to the data.

Both the k-means and SOMs are algorithmic methods and do not have a probabilistic justification. *Probabilistic model-based* clustering (PMC) analysis, on the other hand, assumes that the data is generated by a mixture of underlying probability distributions, and uses the maximum-likelihood method to estimate parameters that define the number of clusters as well as the clusters. Hence, we do not need to specify the number of clusters. Using the probabilistic model, we can even consider covariates while determining the clustering memberships of the genes. However, the model can quickly become complicated as the number of clusters increases. Thus, we must try to use parsimonious models as much as possible. Finally, PMC and k-means are also closely related. In fact, k-means can be interpreted as a parsimonious model of simple independent Gaussians [33.15, 45, 46].

Example 2: Clustering Analysis. In this example, first we perform a hierarchical clustering analysis on the 490 SDE genes from example 1. The clustering analysis is applied in two directions: clustering on samples and clustering on genes. Although we do not present the entire the clustering tree here, two major clusters are formed to distinguish tumor and normal samples. For clustering on the genes, there are roughly five major patterns in terms of the gene expressions. One pattern corresponds to the underexpressed genes and the other four corresponds to the overexpressed genes in the tumor samples versus the normal ones.

For illustration, we selected the first 10 normal arrays and the first 10 cancer arrays, and 20 overexpressed and 20 underexpressed genes randomly from the 490 SDE genes. Figure 33.3 is from the heatmap function in *R*. Though not perfect, two patterns are formed mostly along the line of normal versus tumor tissues. There are roughly five major patterns in terms of expression profiles. Overexpressed and underexpressed genes tend to belong to different clusters. For example, pattern 3 (P3) and pattern 4 (P4) are mainly composed of underexpressed genes while the other three clusters contain mainly overexpressed genes.

Following the hierarchical clustering analysis presented above, we also applied the k-means approach to the 490 SDE genes and set the number of clusters to five. Furthermore, we applied *probabilistic model-based* clustering (PMC) to the same dataset. We examined the BIC (Bayesian information criterion) for different numbers of clusters, and it turned out that the value of

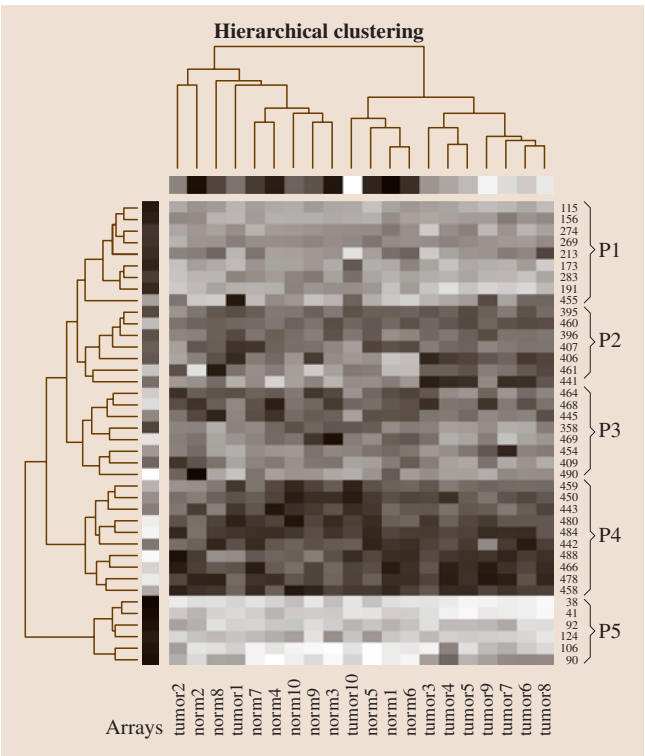

Fig. 33.3 Hierarchical clustering based on a subset of the colon-cancer dataset. Each column corresponds to a sample, and each row a gene. The underexpressed genes were assigned numbers above 440, and the overexpressed genes at or below 440

Table 33.1 The numbers of genes belonging to the intersects of the five k-means clusters and the 13 PMC clusters

k-Means Clusters	PMC Clusters												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	11	0	0	0	57	35	0	0	0	0	29	0
2	25	8	0	0	62	5	0	0	0	0	0	0	0
3	0	0	2	13	0	0	0	0	23	0	0	0	0
4	0	0	15	0	0	0	0	0	1	31	0	0	41
5	0	2	0	0	0	0	1	65	0	34	24	6	0

BIC reaches its minimum at 13 clusters, which is much more than heuristic choice of five. Table 33.1 displays the numbers of genes belonging to the intersects of the five k-means clusters and the 13 PMC clusters. Each of the five k-means clusters is a union of four or so PMC clusters. In fact, if we choose five PMC clusters, they are very similar to the formation of the five k-means clusters, and we will assess this similarity in the next section.

Measure of Agreement Between Two Sets of Clusters

From both the methodological and biologic points of view, there is a need to compare the clusters from different clustering methods. For example, to evaluate the performance of a new clustering approach, we need to compare the derived clusters with the underlying membership in a simulation study. We may also be interested in comparing clustering results derived from the same mRNA samples but being hybridized and analyzed in two different laboratories.

A commonly used measure of agreement between two sets of clusters is the so-called adjusted Rand index (ARI) [33.15, 47, 48]. Let us consider the partitions U and V, and let $\boldsymbol{n}_{ij}$ be the number of genes falling in the intersect of the i-th cluster in U and the j-th cluster in V. The ARI is defined as

$$\frac{\sum_{i,j}\binom{n_{ij}}{2}-\left[\sum_i\binom{n_{i\cdot}}{2}\sum_j\binom{n_{\cdot j}}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_i\binom{n_{i\cdot}}{2}+\sum_j\binom{n_{\cdot j}}{2}\right]-\left[\sum_i\binom{n_{i\cdot}}{2}\sum_j\binom{n_{\cdot j}}{2}\right]/\binom{n}{2}},$$

where $n_{i\cdot}$ and $n_{\cdot j}$ are the numbers of genes in the i-th cluster of U and the j-th cluster of V, respectively.

We suggested some similarity between the k-means and PMC clusters. In fact, the ARI value between the two sets of clusters is 0.425, and it increases to 0.94 if both methods use five clusters. This similarity is expected, because PMC and k-means are equivalent if PMC assumes an independent Gaussian covariance structure [33.15].

33.2.2 Classification

In most microarray experiments, we know the groups on the arrays. For example, some mRNA samples were extracted from tumor cells and the others from normal cells. This is similar to the situation in Sect. 33.1.1. Therefore, it is natural to use this information in analysis and to class cells based on the expression profiles. This is so-called supervised learning.

In Sect. 33.1.1, $Y_{ij,k}$ denotes the expression level of the i-th gene on the j-th array in the k-th sample. Here, we also use $(Y_{ij}, Z = k)$ to reflect the fact that the expression level Y_{ij} of the i-th gene on the j-th array comes from the k-th sample. In other words, the sample group is represented by Z, which is the response or dependent variable in classification.

The essence of classification is to define domains in the feature space spanned by Y_{ij} and to assign a class membership Z to each domain. Classification methods differ in the choice of the shape for the domain and in the algorithm to identify the domain. Some elementary concepts are useful to distinguish these differences. The first one is *linearity*. It refers to a linear combination of the features (expressions of different genes) that forms a hyperplane separating different domains in the feature space. The second term is *separability*. It reflects the extent that the different classes of samples are separable. The third concept is *misclassification*. Often, data are only partially separable, and misclassification is inevitable. In this circumstance, we may need to define a cost function to accommodate different classification errors.

In the machine-learning literature, there is also a distinction between the *learning* (i.e., training) and the *test* samples. The learning data are used to train the classification algorithm and the test data are used to test the predictive ability of the trained classification algorithm. In practice, however, we usually have one dataset and have to split the sample into the training and test samples by leaving a portion of data out during the learning process and saving it as the test data. This pro-

cedure is called cross-validation. More precisely, for a v-fold cross-validation, we first divide the data into v approximately equal sub-samples. Then, we use $v-1$ sub-samples as the training data to construct a classification rule and the left-over subsample as the test data to validate the classification rule. After rotating every sub-sample between training and test data, the performance of the classification rule is assessed through the average in the v runs of validation in the test sample.

In the next subsections, we will review four classification methods that are useful for classifying tissue samples based on gene expression profiles. The methods are linear discriminant analysis (LDA), support vector machines (SVM), artificial neural networks (ANN), and tree-based classification.

LDA

LDA was introduced by Fisher in 1936 for classifying samples by finding a hyperplane that maximizes the between-class variances. Let S_Y be the common sample covariance matrix of all gene expressions, $\bar{Y}_1$ and $\bar{Y}_2$ be the average expression levels of the genes in groups 1 and 2, respectively. The solution to LDA is $S_Y^{-1}(\bar{Y}_1 - \bar{Y}_2)$.

SVM

SVM was first proposed by *Boser* et al. [33.49] and *Cortes* and *Vapnik* [33.50]. SVM finds an optimal hyperplane to separate samples and to allow the maximum *separation* between different classes of samples. The margin of the region that separates samples is *supported* by a few vectors, termed support vectors.

In a two-class classification problem, let $Z = 1$ or -1 denote the two classes. If the two classes of samples are separable, we find a hyperplane $\{y : y^{\mathrm{T}}\beta + \beta_0 = 0, ||\beta|| = 1\}$ such that $(y^{\mathrm{T}}\beta + \beta_0)Z \geq C \geq 0$, where C is the margin optimized to allow the maximal space between the two classes of samples.

For nonseparable case, the procedure is much complicated. Some points will inevitably be on the wrong side of the hyperplane. The idea is to introduce a slack variable to reflect how far a sample is on the wrong side, and then look for the hyperplane at the condition of the total misclassification less than a user-selected limit (i. e., bound the sum of slack variables by a constant). We refer to *Vapnik* [33.51] for the details.

Example 3: Support Vector Machine (SVM). In this example, we perform a classification analysis on the colon-cancer data by SVM. We use M26697 and M63391, the two most significant genes that were identified by SAM from example 1. Specifically, M26697 is the most significant overexpressed gene and M63391 is the most significant underexpressed gene. We used the SVM function in *R* with the cost equal to 100, γ of 1 and tenfold cross-validation, where γ is the coefficient of the radial kernel used to form a hyperplane. Figure 33.4 displays the contour plot of the SVM result. The prediction model correctly classifies 37 cancer and 20 normal samples, but misclassifies three cancer and two normal samples.

Neural Network

The artificial neural network (ANN) is a very popular methodology in machine learning. Also referred to as connectionist architectures, parallel distributed processing, and neuromorphic systems, ANN is an information-processing paradigm with collections of mathematical models that emulate the densely interconnected, parallel structure of the mammalian brain and adaptive biological learning. It is composed of a large number of highly interconnected processing elements that are analogous to neurons and are tied together with

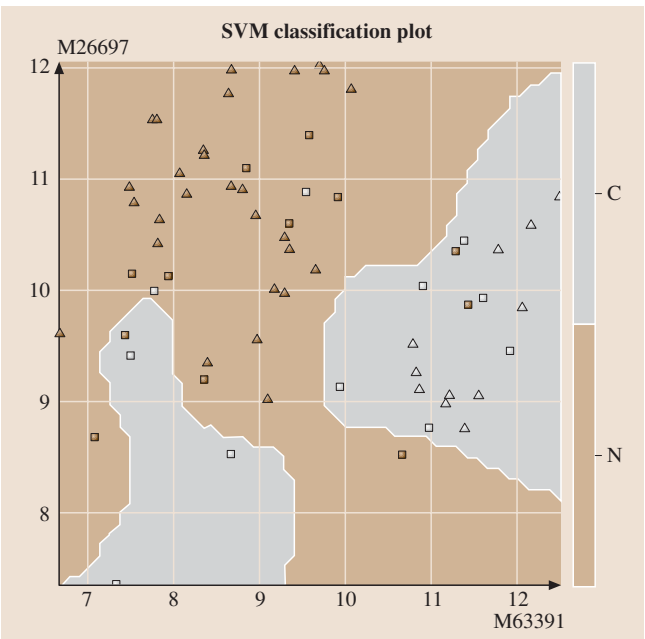

Fig. 33.4 Contour plot of the SVM result using two genes: M26997 and M63391 for the colon-cancer data. C represents cancer and N represents normal. The *light-gray area* is the cancer region and the *brown area* is the normal region. *Square points* represent the support vectors and the *triangle points* represent the data points other than support vectors. The *brown* and *white points* belong to the cancer and the normal regions, respectively

weighted connections that are analogous to synapses. Learning typically occurs by example through training, or exposure to a true set of input/output data where the training algorithm iteratively adjusts the connection weights (synapses). These connection weights store the knowledge necessary to solve specific problems.

ANN can be used for feature selection and feature extraction. The former amounts to variable selection and reduction in statistics and the latter is a generation of the statistical techniques such as principal component analysis, factor analysis, and linear discriminant analysis that are intended to identify lower-dimensional data structures such as linear directions. These lower-dimensional structures usually depend on all of the original variables (i. e., features). Thus, ANN is in essence a computationally intensive version of traditional statistical methods such as regression, classification, clustering, and factor analysis. However, ANN is designed in a way that mimics neural networks and is biologically intuitive and appealing in many applications. This is the major reason that we plan to consider ANN as one of the primary tools to explore the unknown relationship in our data, which is usually referred to as pattern recognition.

The advantage of ANNs lies in their resilience against distortions in the input data and their capability for learning. They are often good at solving problems that are too complex for conventional technologies (e.g., problems that do not have an algorithmic solution, or for which an algorithmic solution is too complex to be found), and are often well-suited to problems that people are good at solving, but for which traditional methods are not.

There are multitudes of different types of ANNs. Some of the more popular include the multilayer perceptron, which is generally trained with the back-propagation of error algorithm, learning vector quantization, radial basis functions, Hopfield, and Kohonen, to name a few. Some ANNs are classified as feed-forward while others are recurrent (i. e., implement feedback) depending on how data is processed through the network. Some ANNs employ supervised training while others are referred to as unsupervised or self-organizing.

Figure 33.5 illustrates a conventional three-layer neural network with n features and K classes. For this feed-forward neural network, the inputs are $y_1, \cdots, y_n$ which correspond to the gene expression profiles and the outputs are $z_1, \cdots, z_K$, which correspond to the K samples in the microarray data. The middle layer consists of many hidden units (also called neurons) and the number of hidden units can be freely chosen and determine the maximum nonlinearity. Each line in Fig. 33.5 indicates a weight—the edge—in the network. This weight represents how much the two neurons which are connected by it can interact. If the weight is larger, then the two neurons can interact more, that is, a stronger signal can pass through the edge. The nature of the interconnections between two neurons can be such that one neuron can either stimulate (a positive weight α) or inhibit (a negative weight α) the other. More precisely, in each hidden unit, we have

$$X_m = \sigma\left(\alpha_{0m} + \alpha_m^{\mathrm{T}} Y\right) ,$$

where σ is called the activation function or neural function and $(\alpha_{0m}, \alpha_m^{\mathrm{T}})$ are the weights. A common choice for σ is the sigmoid function,

$$\sigma(\upsilon) = \frac{1}{1 + \mathrm{e}^{-\upsilon}} .$$

The output function allows a final transformation of the linear combinations of the hidden unit variables,

$$f_k(z) = g_k\left(\beta_{0k} + \beta_k^{\mathrm{T}} X\right) .$$

For a K-class classification, a softmax (logistic) function is usually chosen for the output function

$$g_k(T) = \frac{\mathrm{e}^{T_k}}{\sum_{l=1}^{K} \mathrm{e}^{\mathrm{T}_l}} .$$

During the training period we present the perceptron with inputs one at a time and see what output it gives. If

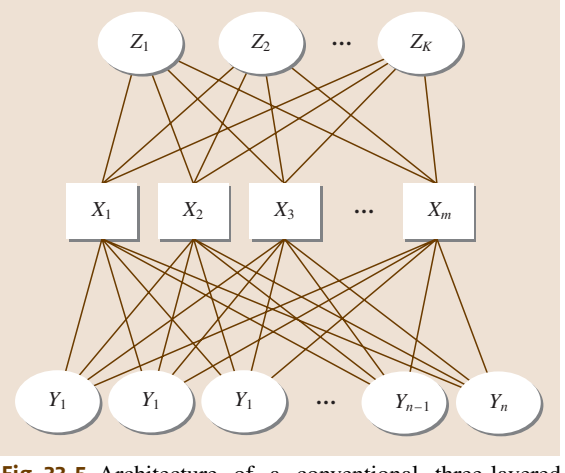

Fig. 33.5 Architecture of a conventional three-layered feed-forward neural network

the output is wrong, we will tell it that it has made a mistake. It should then change its weights and/or threshold properly to avoid making the same mistake later.

33.2.3 Tree- and Forest-Based Classification

One of the most convenient and intuitive approaches for classification is classification trees [33.52, 53]. Classification trees, and their expansion to forests, are based on the so-called recursive partitioning technique. The basic idea of recursive partitioning is to extract homogeneous strata of the tissue samples through expression profiles depending on the expression levels of a particular gene.

Zhang and *Yu* [33.54] reanalyzed the dataset from *Hedenfalk* et al. [33.55] to classify breast cancer mutations in either the BRCA1 or BRCA2 gene using gene expression profiles. *Hedenfalk* et al. [33.55] collected and analyzed biopsy specimens of primary breast cancer tumors from seven and eight patients with germline mutations of BRCA1 and BRCA2, respectively. In addition, seven patients with sporadic cases of primary breast cancer whose family history was unknown were also identified. They obtained cDNA microarrays from 5361 unique genes, of which 2905 are known genes and 2456 are unknown. Thus, in this dataset, Let $Z = 1, 2, 3$ denote BRCA1, BRCA2, and sporadic cases, respectively.

If we use this entire breast cancer dataset to construct a tree, these 22 samples form the initial learning sample, which is called the root node and labeled as node 1 in the tree diagram (Fig. 33.6). The tree structure is determined by recursively selecting a split to divide an upper layer node into two offspring nodes. To do this, we need to evaluate the homogeneity, or the impurity to its opposite,

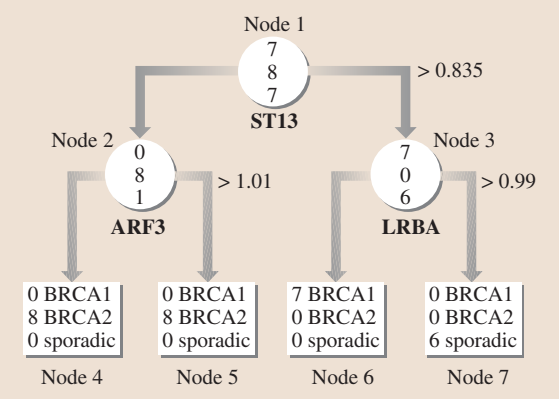

Fig. 33.6 Classification tree for breast-cancer data

of any node. A common measure of node impurity is the entropy function,

$$i_t = -\sum_{k=1}^{K} P(Z = k|\text{node } t) \log[P(Z = k|\text{node } t)] .$$

If node t is the root node, then $P(Z = 1|\text{node } t) = 7/22$, $P(Z = 2|\text{node } t) = 8/22$, and $P(Z = 3|\text{node } t) = 7/22$. Thus, the impurity i_t of the root node can be calculated easily as follows: $i_t = -(7/22)\log(7/22) - (8/22)\log(8/22) - (7/22)\log(7/22) = 1.097$.

How good is the root node? The impurity is zero for a perfect node in which $P(Z = k|\text{Node } t)$ is either 0 or 1, and reaches its worst level when $P(Z = k|\text{node } t) = \frac{1}{3}$ with $i_t = 1.099$. Therefore, the impurity of the root node is near the worst level by design, motivating us to partition the root node into small nodes to reduce the impurity.

The first step of the recursive partitioning process is to divide the root of 32 samples in Fig. 33.6 into two nodes, namely, nodes 2 and 3 in Fig. 33.6. There are many ways of partitioning the root node, because we can take any of the 5361 genes and split the root node according to whether the expression level of this chosen gene is greater than any threshold c. After comparing all possible partitions, we choose the gene and its threshold to keep both i_2 in node 2 and i_3 in node 3 at their lowest possible levels simultaneously. Mathematically, we achieve this goal by minimizing the weighted impurity $r_2 i_2 + r_3 i_3$, where r_2 and r_3 are the proportions of tissue samples in nodes 2 and 3, respectively. This is precisely how the first split (i. e., whether ST13 > 0.835) in Fig. 33.6 is determined.

Once the root is split into nodes 2 and 3, and we can apply the same procedure to potentially split nodes 2 and 3 further. Indeed, the tree in Fig. 33.6 divides the 22 samples into four groups using Heping Zhang's RTREE (http://peace.med.yale.edu). Nodes 2 and 3 are divided based on the expression levels of genes ARF3 and LRBA.

Using a variety of analytic techniques including a modified F- and t-test and a mutual-information scoring, *Hedenfalk* et al. [33.55] selected nine differentially expressed genes to classify BRCA1-mutation-positive and negative tumors and then 11 genes for BRCA2-mutation-positive and negative tumors. Clearly, the tree in Fig. 33.6 uses fewer genes and is a much simpler classification rule.

Although Fig. 33.6 is simple, it does not contain the potentially rich information in the dataset. To improve the reliability of the classification and to accommo-

date potentially multiple biological pathways, *Zhang* and *Yu* [33.54] and *Zhang* et al. [33.56] proposed expanding trees to forests. The large number of genes in microarray makes it an ideal application for these forests.

The most common approach to constructing forests is to perturb the data randomly, form a tree from the perturbed data, and repeat this process to form a series of trees; this is called a random forest. After a forest is formed, we aggregate information from the forest. One such scheme, called bagging (bootstrapping and aggregating), generates a bootstrap sample from the original sample. The final classification is then based on the majority vote of all trees in the forest [33.57].

It is well-known that random forests [33.18, 57] improve predictive power in classification. After observing the fact that there are typically many trees that are of equally high predictive quality in analyzing genomic data, *Zhang* et al. [33.18] proposed a method to construct forests in a deterministic manner. Deterministic forests eliminate the randomness in the random forests and maintain a similar, and sometimes improved, level of precision as the random forests.

The procedure for constructing the deterministic forests is simple. We can search and collect all distinct trees that have a nearly perfect classification or are better than any specified precision. This can be carried out by ranking the trees in deterministic forests. One limitation for the forests (random or deterministic) is that we cannot view all trees in the forests. However, we can examine the frequency of genes as they appear in the forests. Frequent and prominent genes may then be used and analyzed by any method as described above. In other words, forest construction offers a mechanism for data reduction. For the breast-cancer data, one of the most prominent genes identified in the forests is ERBB2. *Kroll* et al. [33.58] analyzed the gene expression patterns of four breast-cancer cell lines: MCF-7, SK-BR-3, T-47D, and BT-474, and reported unique high levels of expressions in the receptor tyrosine kinase ERBB2.

33.3 Fourth-Level Analysis of Microarray Data

Nowadays, different types and platforms of microarray have been developed to address the same (or similar) biological problems. How to integrate and exchange the information contained in different sources of studies effectively is an important and challenging topic for both biologists and statisticians [33.59]. The strategy depends on the situation. When all studies of interest were conducted under the same experimental conditions, this is a standard situation for meta-analysis. There are situations where the experiments are similar, but different platforms were measured, such as the integration of one cDNA array-based study and one oligonucleotide array-based study. There are also situations where different biomolecule microarrays were collected, such as the integration of a genomic array study and a proteomic array study.

Integrating a cDNA array and an Affymetrix chip is complicated because genes on a cDNA array may correspond to several genes (or probesets) on the Affymetrix chip based on the Unigene cluster-matching criteria [33.60]. Instead of matching by genes, matching by the sequence-verified probes may increase the correlation between two studies [33.61].

Most meta-analyses of microarray data have been performed in a study-by-study manner. For example, *Yauk* et al. [33.62] use the Pearson coefficient to measure the correlation across studies, *Rhodes* et al. [33.63] and *Wang* et al. [33.64] use the estimations from one study as prior knowledge while analyzing other studies, and *Welsh* et al. [33.65] treat DNA microarrays as a screening tool and then use protein microarrays to identify the biomarker in cancer research. While they are convenient, these strategies are not ideal [33.63, 66]. Thus, it is imperative and useful to develop better methods to synthesize information from different genomic and proteomic studies [33.59, 62, 67].

33.4 Final Remarks

The technology of gene and protein chips is advancing rapidly, and the entire human genome can be simultaneously monitored on a single chip. The analytic methodology is evolving together with the technology development, but is far from satisfactory. This article reviews some of the commonly used methods in ana-

lyzing microarrays. Analyzing microarray data is still challenging; some of the important issues include how to interpret the results in the biological context, how to improve the reproducibility of the conclusions, and how to integrate information from related but different studies.

References

33.1 M. Schena, M. Shalon, R.W. Davis, P.O. Brown: Quantitative monitoring of gene-expression patterns with a complementary-DNA microarray, Science **270**, 467–470 (1995)

33.2 R.A. Heller, M. Schena, A. Chai, D. Shalon, T. Bedilion, J. Gilmore, D.E. Woolley, R.W. Davis: Discovery and analysis of inflammatory disease-related genes using cDNA microarrays, Proc. Natl. Acad. Sci. USA **94**(6), 2150–2155 (1997)

33.3 E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, N. Friedman: Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data, Nature Genetics **34**, 166–176 (2003)

33.4 J.C. Hacia, B. Sun, N. Hunt, K. Edgemon, D. Mosbrook, C. Robbins, S.P.A. Fodor, D.A. Tagle, F.S. Collins: Strategies for mutational analysis of the large multiexon ATM gene using high-density oligonucleotide arrays, Genome Res. **8**, 1245–1258 (1998)

33.5 J.B. Fan, X.Q. Chen, M.K. Halushka, A. Berno, X.H. Huang, T. Ryder, R.J. Lipshutz, D.J. Lockhart, A. Chakravarti: Parallel genotyping of human SNPs using generic high-density oligonucleotide tag arrays, Gen. Res. **10**, 853–860 (2000)

33.6 S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C.H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J.P. Mesirov, T. Poggio, W. Gerald, M. Loda, E.S. Lander, T.R. Golub: Multiclass cancer diagnosis using tumor gene expression signatures, Proc. Natl. Acad. Sci. USA **98**, 15149–15154 (2001)

33.7 E.R. Marcotte, L.K. Srivastava, R. Quirion: DNA microarrays in neuropsychopharmacology, Trends Pharmacol. Sci. **22**, 426–436 (2001)

33.8 C. Li, W.H. Wong: Model-based analysis of oligonucleotide arrays: expression index computation, outlier detection, Proc. Natl. Acad. Sci. USA **98**, 31–36 (2001)

33.9 B. Efron, R. Tibshirani, J.D. Storey, V. Tusher: J. Amer. Stat. Assoc **96**, 1151–1160 (2001)

33.10 V.G. Tusher, R. Tibshirani, G. Chu: Significance analysis of microarrays applied to the ionizing radiation response, Proc. Natl. Acad. Sci. USA **98**, 5116–5121 (2001)

33.11 R.A. Irizarry, B. Hobbs, F. Collin, Y.D. Beazer-Barclay, K.J. Antonellis, U. Scherf, T.P. Speed: Exploration, normalization, and summaries of high density oligonucleotide array probe level data, Biostat. **4**, 249–264 (2003)

33.12 M.B. Eisen, P.T. Spellman, P.O. Brown, D. Botstein: Cluster analysis and display of genome-wide expression patterns, Proc. Natl. Acad. Sci. USA **95**, 14863–14868 (1998)

33.13 A. Soukas, P. Cohen, N.D. Socci, J.M. Friedman: Leptin-specific patterns of gene expression in white adipose tissue, Genes Dev. **14**(8), 963–980 (2000)

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

33.15 K.Y. Yeung, W.L. Ruzzo: Principal component analysis for clustering gene expression data, Bioinformatics **17**, 763–774 (2001)

33.16 K.Y. Yeung, C. Fraley, A. Murua, A.E. Raftery, W.L. Ruzzo: Model-based clustering and data transformations for gene expression data, Bioinformatics **17**, 977–987 (2001)

33.17 O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, R.B. Altman: Missing value estimation methods for DNA microarrays, Bioinformatics **17**(6), 520–525 (2001)

33.18 H.P. Zhang, C. Yu, B. Singer: Cell and tumor classification using gene expression data: construction of forests, Proc. Natl. Acad. Sci. USA **100**, 4168–4172 (2003)

33.19 T.S. Furey, N. Cristianini, N. Duffy, D.W. Bednarski, M. Schummer, D. Haussler: Support vector machine classification and validation of cancer tissue samples using microarray expression data, Bioinformatics **16**(10), 906–914 (2000)

33.20 K. Mehrotra, C.K. Mohan, S. Ranka: *Elements of Artificial Neural Networks* (MIT, Massachusetts 1997)

33.21 H.P. Zhang, C. Yu, B. Singer, M. Xiong: Recursive partitioning for tumor classification with gene expression microarray data, Proc. Natl. Acad. Sci. USA **98**, 6730–6735 (2001)

33.22 A.J. Butte, P. Tamayo, D. Slonim, T.R. Golub, I.S. Kohane: Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks, Proc. Natl. Acad. Sci. USA **97**, 12182–12186 (2000)

33.23 P. D'haeseleer, S. Liang, R. Somogyi: *Gene expression data analysis and modeling* (Pacific Symposium on Biocomputing, 1999)

33.24 I. Shmulevich, E.R. Dougherty, S. Kim, W. Zhang: Probabilistic Boolean networks: a rule-based un-

certainty model for gene regulatory networks, Bioinformatics **18**(2), 261–274 (2002)
33.25 N. Friedman, M. Linial, I. Nachman, D. Pe'er: Using Bayesian networks to analyze expression data, J. Comp. Biol. **7**, 601–620 (2000)
33.26 E. Segal, B. Taskar, A. Gasch, N. Friedman, D. Koller: Rich probabilistic models for gene expression, Bioinformatics **1**, 1–10 (2001)
33.27 D.J. Lockhart, H. Dong, M.C. Byrne, M.T. Follettie, M.V. Gallo, M.S. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Horton, E.L. Brown: Expression monitoring by hybridization to high-density oligonucleotide arrays, Nat. Biotechnol. **14**, 1675–1680 (1996)
33.28 G. Smyth: Linear models and empirical Bayes methods for assessing differential expression in microarray experiments, Stat. Appl. Genet. Mol. Biol, **3**(1), 3 (2004)
33.29 Z. Šidák: Rectangular confidence regions for the means of multivariate normal distributions, J. Am. Stat. Assoc. **62**, 626–633 (1967)
33.30 S. Draghici: *Data analysis tools for DNA microarrays* (Chapman, Hall/CRC, New York 2003)
33.31 Y. Benjamin, Y. Hochberg: Controlling the false discovery rate – a practical and powerful approach to multiple testing, J. Roy. Soc. B Met. **57**(1), 289–300 (1995)
33.32 J.D. Storey: A direct approach to false discovery rates, J. R. Stat. Ser. B Stat. Methodol. **64**, 479–498 Part 3 (2002)
33.33 J.D. Storey: A Bayesian interpretation, the q-value, Ann. Stat, **31**(6), 2013–2035 (2003)
33.34 J.F. Troendle: Stepwise normal theory multiple test procedures controlling the false discovery rate, J. Stat. Plan. Inference **84**(1-2), 139–158 (2000)
33.35 B. Efron, R. Tibshirani: Empirical bayes methods and false discovery rates for microarrays, Genet. Epidemiol. **23**(1), 70–86 (2002)
33.36 I. Lonnstedt, T. Speed: Replicated microarray data, Stat. Sinica **12**(1), 31–46 (2001)
33.37 U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, A.J. Levine: Broad patterns of gene expression revealed by clustering analysis of tumor, normal colon tissues probed by oligonucleotide arrays, Proc. Natl. Acad. Sci. USA **96**, 6745–6750 (1999)
33.38 J. Quackenbush: Computational analysis of microarray analysis, Nature Rev. Genetics **2**, 418–427 (2001)
33.39 N. Kaminski, N. Friedman: Practical approaches to analyzing results of microarray experiments, Am. J. Respir. Cell. Mol. Biol. **27**(2), 125–132 (2002)
33.40 R. Jansen, D. Greenbaum, M. Gerstein: Relating whole-genome expression data with protein-protein interactions, Genome Res. **12**(1), 37–46 (2002)
33.41 J.C. Boldrick, A.A. Alizadeh, M. Diehn, S. Dudoit, C.L. Liu, C.E. Belcher, D. Botstein, L.M. Staudt, P.O. Brown, D.A. Relman: Stereotyped and specific gene expression programs in human innate immune responses to bacteria, Proc. Natl. Acad. Sci. USA **99**, 972–977 (2002)
33.42 G. Sherlock: Analysis of large-scale gene expression data, Curr. Opin. Immunol. **12**(2), 201–205 (2000)
33.43 F.H. Duan, H.P. Zhang: Correcting the loss of cell-cycle synchrony in clustering analysis of microarray data using weights, Bioinformatics **20**(11), 1766–1771 (2004)
33.44 T. Kohonen: *Self-Organizing Maps* (Springer, Brelin Heidelberg New York 1997)
33.45 W.N. Venables, B.D. Ripley: *Modern Applied Statistics with S* (Springer, Berlin Heidelberg New York 2002)
33.46 E. Wit, J. McClure: *Statistics for Microarrays* (Wiley, New York 2004)
33.47 L. Hubert, P. Arabie: Comparing partitions, J. Classification **2**, 193–218 (1985)
33.48 G.W. Milligan, M.C. Cooper: A study of the comparability of external criteria for hierarchical cluster-analysis, Multivairate Behavioral Research **21**(4), 441–458 (1986)
33.49 B.E. Boser, I.M. Guyon, V.N. Vapnik: A training algorithm for optimal margin classifiers. In: *Fifth Annual Workshop on Computational Learning Theory*, ed. by D. Haussle (ACM, New York 1992) pp. 144–152
33.50 C. Cortes, V. Vapnik: Support-vector networks, Mach. Learn. **20**(3), 273–297 (1995)
33.51 V. Vapnik: *Statistical Learning Theory* (Wiley, New York 1998)
33.52 L. Breiman, J. Friedman, C. Stone, R. Olshen: *Classification, Regression Trees* (Wadsworth, Belmont 1984)
33.53 H.P. Zhang, B. Singer: *Recursive Partitioning in the Health Sciences* (Springer, Berlin Heidelberg New York 1999)
33.54 H. Zhang, C.-Y. Yu: Tree-based analysis of microarray data for classifying breast cancer, Front. in Biosci. **7**, c63–67 (2002)
33.55 I. Hedenfalk, D. Duggan, Y. Chen, M. Radmacher, M. Bittner, R. Simon, P. Meltzer, B. Gusterson, M. Esteller, M. Raffeld, Z. Yakhini, A. Ben-Dor, E. Dougherty, J. Kononen, L. Bubendorf, W. Fehrle, S. Pittaluga, S. Gruvberger, N. Loman, O. Johannsson, H. Olsson, B. Wilfond, G. Sauter, O.P. Kallioniemi, A. Borg, J. Trent: Gene-expression profiles in hereditary breast cancer, N. Engl. J. Med **344**, 539–48 (2001)
33.56 H.P. Zhang, C.Y. Yu, H.T. Zhu, J. Shi: Identification of linear directions in multivariate adaptive spline models, J. Am. Stat. Assoc. **98**, 369–376 (2003)
33.57 B.L. Random: Random forests, Mach. Learn. **45**, 5–32 (2001)

33.58 T. Kroll, L. Odyvanova, H. Clement, C. Platzer, A. Naumann, N. Marr, K. Hoffken, S. Wolfl: Molecular characterization of breast cancer cell lines by expression profiling, J. Cancer Res. Clin. Oncol. **128**, 125–34 (2002)

33.59 Y. Moreau, S. Aerts, B. D. Moor, B. D. Strooper, M. Dabrowski: Comparison and meta-analysis of microarray data: from the bench to the computer desk, Trends Genetics **9**(10), 570–577 (2003)

33.60 D. Ghosh, T. Barette, D. Rhodes, A. Chinnaiyan: Statistical issues and methods for meta-analysis of microarray data: a case study in prostate cancer, Funct. Integrat. Gen. **3**(4), 180–188 (2003)

33.61 B. H. Mecham, G. T. Klus, J. Strover, M. Augustus, D. Byrne, P. Bozso, D. Z. Wetmore, T. J. Mariani, I. S. Kohane, Z. Szallasi: Sequence-matched robes produce increased cross-platform consistency and more reproducible biological results in microarray-based gene expression measurements, Nucleotide Acids Res. **32**(9), e74 (2004)

33.62 C. L. Yauk, M. L. Berndt, A. Williams, G. R. Douglas: Comprehensive comparison of six microarray technologies, Nucleic Acids Res. **32**(15), e124 (2004)

33.63 D. R. Rhodes, T. R. Barrette, M. A. Rubin, D. Ghosh, A. M. Chinnaiyan: Meta-analysis of microarrays: interstudy validation of gene expression profiles reveals pathway dysregulation in prostate cancer, Cancer Res. **62**(15), 4427–4433 (2002)

33.64 J. Wang, K. R. Coombes, W. E. Highsmith, M. J. Keating, L. V. Abruzzo: Differences in gene expression between B-cell chronic lymphocytic leukemia and normal B cells, Bioinformatics **20**(17), 3166–3178 (2004)

33.65 J. B. Welsh, L. M. Sapinoso, S. G. Kern, D. A. Brown, T. Liu, A. R. Bauskin, R. L. Ward, N. J. Hawkins, D. I. Quinn, P. J. Russell, R. L. Sutherland, S. N. Breit, C. A. Moskaluk, H. F. Frierson Jr., G. M. Hampton: Large-scale delineation of secreted protein biomarkers overexpressed in cancer tissue and serum, Proc. Natl. Acad. Sci **100**(6), 3410–3415 (2003)

33.66 L. V. Hedges, I. Olkin: *Statistical Methods For Meta-Analysis* (Academic, New York 1985)

33.67 A. K. Järvinena, S. Hautaniemib, H. Edgrena, P. Auvinend, J. Saarelaa, O. P. Kallioniemic, O. Monni: Are data from different gene expression microarray platforms comparable?, Genomics **83**(6), 1164–1168 (2004)