Journal of Pathology
J Pathol 2001; **195**: 53–65.
DOI: 10.1002/path.891

Review Article

Analysing gene expression data from DNA microarrays to identify candidate genes

Thomas D. Wu*
Department of Bioinformatics, Genentech, Inc., South San Francisco, CA 94080, USA

*Correspondence to:
Thomas D. Wu, MD, PhD,
Genentech, Inc., 1 DNA Way
MS 93, South San Francisco,
CA 94080, USA.
E-mail: twu@gene.com

Abstract

Microarray data analysis can be divided into two tasks: grouping of genes to discover broad patterns of biological behaviour, and filtering of genes to identify specific genes of interest. Whereas the gene-grouping task is largely addressed by cluster analysis, the gene-filtering task relies primarily on hypothesis testing. This review article surveys analytical methods for the gene-filtering task. Various types of data analysis are discussed for four basic types of experimental protocols: a comparison of two biological samples; a comparison of two biological conditions; each represented by a set of replicate samples; a comparison of multiple biological conditions; and analysis of covariate information. Copyright © 2001 John Wiley & Sons, Ltd.

Keywords: DNA microarrays; gene expression; bioinformatics; functional genomics; data analysis; review article

Introduction

Genomics research has transformed molecular biology from a data-poor to a data-rich science. Genomics data initially came from several large-scale sequencing efforts, including the Human Genome Project [1]. Now attention has shifted from sequencing towards the identification of gene function. A primary tool for this task is the gene expression microarray, colloquially referred to as a 'gene chip' [2]. Microarrays are rapidly becoming standard laboratory tools [3], but because they generate large volumes of data, they necessitate new computational and statistical techniques to manage and analyse gene expression data.

The fields of bioinformatics and computational biology have emerged to address the computational needs of genomics research. Microarray experiments in particular have raised a wide range of computational requirements, including image processing [4], instrumentation and robotics [5], database design [6,7], data storage and retrieval [8], microarray design based on available expressed sequence tags (ESTs) [9], and data analysis [10]. Furthermore, microarray data need to be interpreted in the context of other biological knowledge, involving various types of 'post-genomics' informatics [11], including gene networks [12], gene pathways [13], and gene ontologies [14].

In this paper, we focus on microarray data analysis and in particular, on methods for identifying particular genes of interest. We make a broad distinction between two types of analysis tasks: gene grouping and gene filtering. Early analyses of microarray data focused on reducing the complexity of the data, by clustering genes into groups [15,16]. This type of analysis has proven useful for providing a general view of the basic biological processes in a given cell or tissue. Sets of co-regulated genes can also be subjected to further analysis; for example, to detect possible transcription factor binding sites [17–20].

In contrast, in gene filtering, we are not interested in understanding the entire genome, but rather in identifying specific genes that are expressed differentially in one or more biological conditions. This type of data analysis is more akin to hypothesis testing than cluster analysis. Microarray experiments allow us to test the expression of thousands of genes simultaneously and to identify genes of interest. The gene-filtering approach is particularly relevant to drug discovery and development [21–23]. For example, at Genentech, my colleagues and I are interested in finding genes that are overexpressed in various types of cancer. Likewise, microarrays are being used widely to study several pathological conditions, including breast cancer [24,25], colon cancer [26], inflammation [27], atherosclerosis [28], and pulmonary fibrosis [29].

Gene grouping and gene filtering represent different approaches to analysing microarray data. Gene grouping is useful for understanding common expression patterns, whereas gene filtering is useful for identifying unusual patterns of expression. Because there already exist several review articles on the former approach for microarray data (e.g. Brazma and Vilo [30]), we will focus our attention in this review instead on methods for identifying candidate genes from microarray data.

The different methods for identifying candidate genes depend largely on the type of data available. We will therefore organize our discussion around four basic types of experimental protocols: a comparison of two biological samples; a comparison of two biological conditions, each represented by a set of replicate samples; a comparison of multiple biological conditions; and analysis of covariate information. By

'biological condition', we mean the cell or tissue type or variant, plus the environmental or experimental variable that a given sample represents. The environmental or experimental variable may include temperature; exposure to some stimulus, insult, or treatment; or elapsed time from that exposure. These variables may define groups implicitly, or can be defined explicitly as covariates.

Although biological experiments vary considerably in their design, the data generated by microarray experiments can be viewed as a matrix of expression levels, organized by samples versus genes, as shown in Figure 1. Each sample represents a separate microarray hybridization and generates a set of M expression levels, one for each gene. We call this set of expression levels an *expression signature*, although the term *expression fingerprint* has also been used. In an analysis, we may consider N such samples. For each gene, we can consider its set of expression levels across the different samples, called its *expression profile*. Outside this matrix of expression levels, we may have covariate information for samples, genes, or both. The goal of microarray data analysis is to make inferences among samples, genes, and their expression levels and covariates.

Microarray technologies

Before we discuss the various techniques for microarray data analysis, we should examine the two types of microarray technologies currently being used. Some of the details underlying microarray fabrication have implications for further data analysis, especially because we are not measuring gene expression or even messenger RNA (mRNA) concentrations per se, but rather an intensity level that depends on several factors.

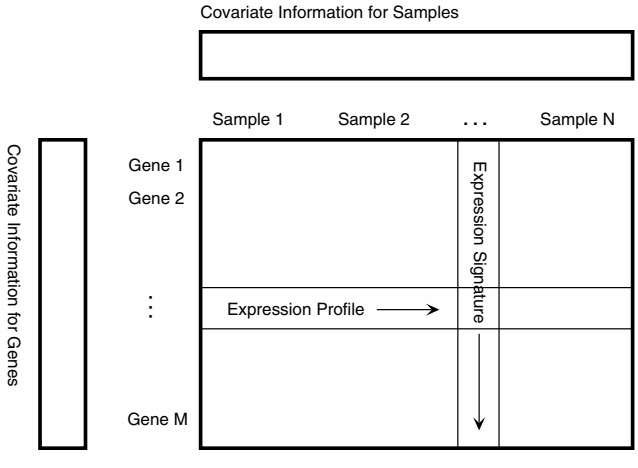

Figure 1. Gene expression data from microarray experiments. Gene expression data may be represented as a matrix of expression levels, organized by samples versus genes. The set of expression levels for a given sample is called an expression signature. The set of expression levels for a given gene is called an expression profile. Data outside this matrix represent covariate information for both samples and genes

Microarrays measure mRNA concentrations by labelling the sample with a dye and then allowing it to hybridize to spots on the array. Each spot contains either DNA oligomers, or a longer DNA sequence designed to be complementary to a particular mRNA of interest. The choice of spotting oligomers or a longer cDNA sequence yields two different microarray technologies: oligo and cDNA microarrays, respectively. Oligo arrays are generated by photolithography techniques to synthesize oligomers directly on the glass slide [31,32]; these arrays are manufactured and marketed primarily by Affymetrix Inc. In contrast, cDNA arrays are created by mechanical gridding, where prepared material is applied to each spot by ink-jet or physical deposition [33,34].

There is generally a one-to-one correspondence between spots and genes, but various exceptions hold. Multiple genes may hybridize to the same spot if the DNA at that spot is not unique to a single gene; this problem is called cross-hybridization. Likewise, a gene may hybridize to more than one spot on a microarray if different spots cover different regions of the gene. In fact, many microarrays are designed deliberately to identify individual exons of a gene, in order to study expression patterns for different splice forms or transcripts. Because of these considerations, it is more accurate to say that each spot measures one or more transcripts of a gene, rather than to a particular gene. Nevertheless, to simplify our discussion in this review, we will ignore this distinction and state that we are measuring expression levels of genes rather than transcripts.

Because cDNA sequences on a microarray are hundreds of nucleotides long, a single spot is usually sufficient to identify a particular gene. However, oligo microarrays have spots that contain oligomers of 25 or so nucleotides. Because such short oligomers will frequently cross-hybridize with several genes, oligo arrays must measure each gene with several oligomers (16–20 in the Affymetrix arrays). Each set of oligomers is called a probe set. A gene is considered present only when the vast majority of the probe set shows positive hybridization.

Oligo microarrays also have another special feature, designed to account for the fact that short oligomers can have non-specific binding and can vary in their hybridization efficiency. Each oligomer on the array has a mismatch oligomer which is intended to serve as a control. The mismatch oligomer is the same as its corresponding perfect match oligomer except for one position (for Affymetrix arrays, the 13th out of 25 positions), which is designed to be different. The amount of specific hybridization can then be measured by taking the difference in hybridization between the perfect match and its corresponding mismatch.

The amount of hybridization to each spot is measured by the intensity of phosphorescent dye at each spot. mRNA samples are labelled with a dye

before hybridization and the non-hybridized samples are then washed off. The remaining dye-labelled and hybridized mRNA is then measured by a camera, which records an intensity level. Each spot may in fact have a gradation of dye intensity, or artefacts arising from dust and other imperfections; these problems are handled by image processing software. More than one sample may be applied to a single microarray, with the different samples being labelled with differently coloured dyes. In practice, however, Affymetrix oligo arrays measure a single sample at a time and therefore use a single type of dye. In contrast, cDNA microarrays measure either one sample or, more commonly, two samples.

Analysis of two samples

The simplest analysis in practice involves two samples, representing a test condition and a control condition. For one-sample arrays, including Affymetrix oligo arrays, the two-sample comparison must be performed computationally, by extracting results from two separate arrays. For cDNA arrays that have been hybridized with two samples, we can obtain data from a single microarray.

Many microarray experiments are designed as two-sample comparisons. For example, microarrays have been used to identify differences in yeast gene expression before and after treatment with various kinase inhibitors [22]. In these experiments, researchers identified candidate genes as those where the expression level increased two-fold from the control sample to the test sample. Although such an analysis would seem to be relatively straightforward, two-sample experiments present several issues that are problematic for data analysis. These issues can greatly affect the accuracy and robustness of two-sample inferences. We discuss three of these issues here, all of which also apply to more complex data sets.

Bias correction

A two-sample analysis essentially yields a list of paired expression values, one pair for each gene. As illustrated in Figure 2, these pairs can be represented graphically by a scatterplot, with the values of sample 1 plotted on the x-axis and the values of sample 2 plotted on the y-axis. In this plot, genes with similar expression levels in the two samples should have points on the identity line, $y=x$, and genes that are expressed differentially should lie at some distance from this line. Genes that are expressed higher in the test sample relative to the control sample will lie above the line and, conversely, genes expressed higher in the control will lie below the line.

However, the problem is that we have not measured expression levels directly, but rather intensity levels, as represented by the amount of phosphorescent dye that was recorded by a camera. Many factors enter into the process from the actual expression levels to the observed intensity levels. For instance, the two samples

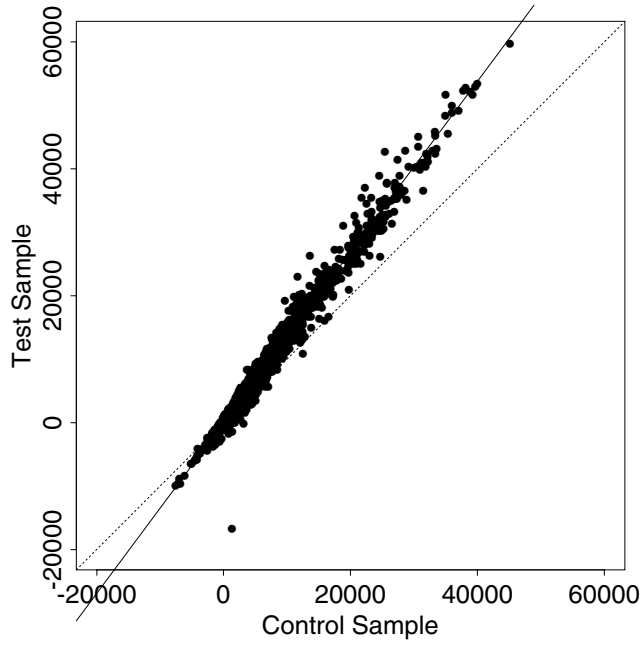

Figure 2. Intensity data from a two-sample experiment. The data are shown as a scatterplot of control intensity levels versus test intensity levels. These data were taken from Affymetrix rat microarrays for two samples of rat heart tissue. The identity line, $y=x$, is shown as a dotted line. The solid line represents the normalization line with slope 1.35, as fitted by regression of y on x

may have different concentrations of mRNA overall, which would cause the intensity from one sample to be a multiplicative factor of the corresponding intensity from the other sample. Consequently, genes with identical expression levels should lie on the line $y=\beta x$, where the slope β would represent the concentration ratio. Other factors – including the concentration, brightness, and relative binding affinity of the dye labels; the exposure time; and the camera sensitivity – should also theoretically produce a multiplicative effect, causing the expression levels to lie on some sloped line $y=\beta x$.

In order to correct differences in intensity levels, we need to perform normalization, or bias correction. We would like to find some normalization curve that can equate the intensity levels of the two samples. For a multiplicative effect, this process is equivalent to finding the straight line that best fits the data. For example, in Figure 2, we use a least-squares regression of the test data on the control data to obtain a line with slope 1.35. In general, though, this method will underestimate the slope somewhat, because a regression of Y on X assumes that the X data are fixed and that errors occur only vertically for the Y data. Likewise, a regression of X on Y would overestimate the slope. To allow for errors in both test data and control data, the most accurate way to compute the line is by using principal components analysis [35]. Applying principal components analysis to our particular data, though, yields only a slight correction: a slope of 1.36. Once we have a normalization curve, we

can adjust the data so that their intensity values are on the same scale. For a slope of β, we would multiply the control data by $\sqrt{\beta}$ and divide the test data by the same factor.

Often, a simple multiplicative model is inadequate. Such cases can be identified on the scatterplot when the data have a curved shape and a straight line fits the data poorly. In such cases, the relationship between intensity values in the two samples is not purely multiplicative. In fact, microarray data may frequently have additive effects in addition to the multiplicative effect. A typical additive effect might be a background intensity level that raises the intensity of all spots on one sample by some amount.

In addition to such additive effects, there are several factors that can affect intensity values non-linearly. For example, there are saturation effects in the hybridization process, where extremely high concentrations of a given mRNA cannot be represented with proportionately high intensities, because there are a limited number of cDNA or oligomer molecules on a given spot available for hybridization. This effect becomes more prominent as we try to increase the sensitivity of microarrays by amplifying the mRNA concentrations in the given samples. Phosphorescent dyes can also exhibit a non-linear quenching effect, because the the amount of dye that can be bound by an mRNA molecule is limited. Finally, cameras themselves can measure intensity levels non-linearly and they typically exhibit saturation at high intensity levels.

Handling such non-linear effects is currently a research issue. Thomas Kepler and his colleagues at the Santa Fe Institute (unpublished technical report 00-09-055) are exploring the use of local regression and smoothing splines to normalize intensity values. Their studies suggest that data with both multiplicative and additive errors can lead to a high rate of false-positive results if only a straight line is used to normalize the data.

We need not use all of the data points to compute a normalization curve. Some genes, such as constitutively expressed 'housekeeping' genes, are more likely to be expressed equally in two given samples and these genes could be used to compute the normalization constant. Other microarray experiments [36] have 'spiked' in a controlled amount of foreign mRNA to each sample to use for normalization. Finally, we can use a computational approach to weight each gene according to the likelihood that it is identically expressed in the two samples. We can compute these weights iteratively, by using an initial normalization curve to compute the likelihood of identical expression and then using the resulting weights to refine the normalization curve. This process, a version of expectation-maximization [37], continues until the normalization curve converges.

Relative expression levels

After we have normalized our data, we can evaluate each gene according to its degree of differential expression. Since each gene is represented by a pair of expression values, we may evaluate them according to their difference or ratio. For cDNA arrays, the standard has been to use the ratio as the measure of relative expression. In contrast, for oligo microarrays, the standard has been to use the difference as the measure, specifically the 'relative average difference' value computed by Affymetrix software. The Affymetrix software actually derives this value from an entire probe set, containing 16 or more perfect match–mismatch pairs of oligomers. The Affymetrix software can also convert each probe set into a log average ratio value, which can be transformed into a ratio, but in practice these ratios are not used as often as the difference values.

In either case, we are interested in those genes that have extreme values for relative expression. We may then set some threshold for selecting genes. For example, one proposed rule [22,36] is to select genes whose ratio exceeds 2.0. However, selecting such a threshold seems somewhat arbitrary, especially because some data sets show more variability than others. Accordingly, some researchers have suggested that the threshold be chosen based on the observed data [4]. Housekeeping genes have also been used to assess the amount of variability; for example, one study [38] set the threshold to be three standard deviations of the expression levels observed in a set of 90 housekeeping genes.

Because the significance of ratio and difference values varies from experiment to experiment, we can attempt to interpret them uniformly by converting them into p values. We can then select genes based on a well-defined and well-understood p-value threshold. The p value could be computed easily if we knew the underlying distribution for relative expression levels under some null hypothesis. Unfortunately, we usually do not know in advance what the underlying distribution is. We might adopt a relatively simple model; namely, that the difference data are distributed according to a Gaussian (or normal) distribution, with some mean μ and standard deviation σ. For ratio data, we would assume that the logarithm of the ratio follows a Gaussian distribution, because log ratios can range from negative infinity to positive infinity, whereas ratios are always positive.

In order to test the assumption of normality, we can use a graphical method called a normal probability plot, where the observed distribution of values is plotted against a normal distribution [39]. This type of plot shows on the y-axis the ordered values for relative expression, and on the x-axis the corresponding unit normal quantiles. The unit normal quantile is the percentile that would be observed if the data came from a normal distribution with mean 0 and standard deviation 1. If the data are well approximated by a normal distribution, we should observe a straight line, where the y-intercept indicates the mean μ and the slope indicates the standard deviation, σ, of the distribution. For instance, in Figure 3, we use a

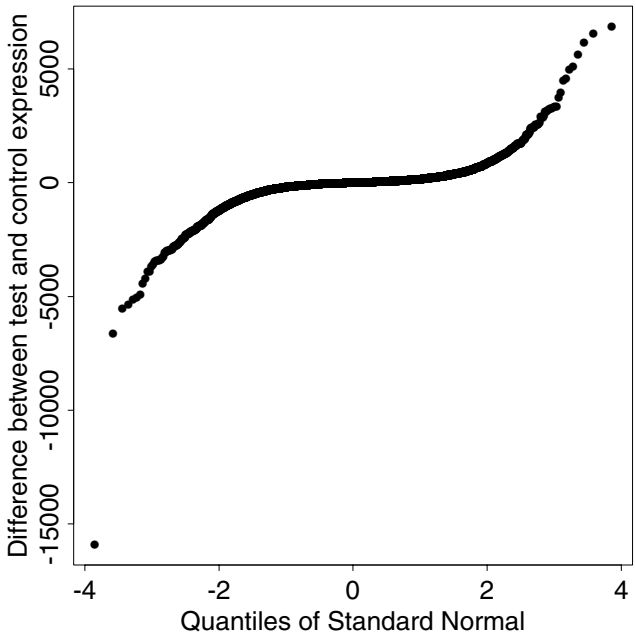

Figure 3. Normal probability plot for microarray data. This plot tests whether the normalized data from Figure 2 fit a normal distribution. Normally distributed data should be represented by a straight line. The curved line shown here indicates that extreme values are larger than they should be for a normal distribution

normal probability plot to evaluate our normalized data from Figure 2. As the figure shows, a straight line does not fit the data well, with the points bending substantially down at the left and up at the right. This pattern indicates that the tails of the distribution are 'heavier' than those for a normal distribution. In other words, the extreme observations are larger than they would be for a normal distribution. This plot indicates either that the null hypothesis is valid and that the extreme values are all significant, or, more likely, that the normal distribution is a poor choice.

In such cases, we cannot use the normal distribution to assign p values. Researchers in bioinformatics and biostatistics are currently investigating various alternatives. One alternative would be to assume a more general type of distribution, such as a gamma distribution, and see whether that distribution fits the data well. Another possibility is to use resampling methods, such as the bootstrap [40], to estimate the underlying distribution from the observed data. Another method for assigning probability values to relative expression values is to use a mixture model [41,42]. A mixture model assumes that there are two or more underlying distributions that are generating the data, and tries to determine both the set of distributions and the appropriate assignment of each data point to a distribution.

Multiple hypothesis testing

Because microarrays allow us to study thousands of genes simultaneously, we are often faced with the problem of multiple hypothesis testing [43]. If we test a statistical method with large volumes of data, even if they are generated randomly, the sheer number of tests should produce a fraction of p values that are low enough to be statistically significant. For example, if we select a p threshold of $\alpha = 0.01$, even a set of random data satisfying the null hypothesis will result in one false-positive per every 100 genes tested. A microarray containing thousands of genes might therefore generate dozens of false-positive results.

One way to handle this problem is to modify the p threshold to account for the number of tests performed. The Bonferroni correction does this by dividing the threshold α by the number of tests performed. For a microarray containing M genes, we would therefore choose only those results that had p values less than α/M. The Bonferroni correction is a somewhat conservative approach to the problem of multiple hypothesis testing. The corrected p threshold ensures that we achieve a false-positive rate of α over the entire set of genes, but arguably sets a criterion that is too strict for each individual gene.

Accordingly, several alternatives to the Bonferroni correction have been suggested [43,44]. One theme underlying these new methods is to rank the observed p values and to apply a different threshold for each p value. The result is that we compare the smallest observed p value against the strictest threshold, but we compare the remaining p values against successively more relaxed thresholds.

Another novel approach is to apply resampling methods, such as the bootstrap [40,45]. A bootstrap sample is an artificial dataset generated from the original data. If the original dataset has M elements, we can generate a bootstrap sample by selecting M elements from that dataset with replacement, which means that after each selection, we allow that element to be chosen again in subsequent selections. In the bootstrap method, we generate several bootstrap samples and record the most extreme statistic T for each bootstrap sample. We can then use the distribution of these extreme statistics to assign a p value to the most extreme value of T that was observed for some gene. If this p value is smaller than our threshold α, we then remove the corresponding gene from the dataset and then repeat the bootstrap process for the remaining genes.

Analysis of two conditions with replicates

Scientific experiments are commonly replicated in order to mitigate the effect of experimental error. Microarray experiments can also benefit from replicate samples to reduce the effect of random fluctuations or noise. In repeating microarray experiments, we can choose either to resample a single cell type or tissue, or to sample from similar cell types or tissues. The former strategy mitigates the problem of 'chip noise', or fluctuations due solely to variations in microarray production and their hybridization. The latter strategy

mitigates the problem of 'biological noise', or fluctuations due to variability across different biological samples.

A recent study of chip noise indicates that there may be substantial variability between microarray experiments, even when samples are taken from the same source [42]. In fact, this study examined not merely chip-to-chip variation, but intra-chip variation by applying a single sample to a special cDNA array with 288 spots printed in triplicate at three locations on the same slide. The authors designed the experiment so that exactly 32 of the 288 spots should be expressed. Their analysis of the three replicates showed that 55, 36, and 58 of the spots appeared to be expressed and that there was considerable inconsistency among the three replicates.

As substantial as chip noise appears to be, biological noise is likely to be even greater. One recent study [46] estimated spot-to-spot, slide-to-slide, and animal-to-animal variability for mouse liver tissues. This study measured spot-to-spot variability by looking at replicated spots on the same slide and found the coefficient of variation (standard deviation divided by the mean) to be 8–18%, depending on the particular gene. The slide-to-slide variability was similar at 15%. But the animal-to-animal variability was greater, ranging from 18–60%, depending on the particular gene.

Many of the issues that we discussed in the two-sample case, such as bias correction, remain important for replicate experiments, although we will not discuss them further. Often the two-sample methods can be generalized to handle replicate experiments. For example, we can extend the methods for bias correction by normalizing across a series of N samples, rather than one sample against another. In this case, the solution involves fitting a normalization curve or line in N-dimensional space.

Comparison of replicate samples

Replicate samples for two conditions may be compared by using the t-test [39]. The t-test measures the difference between the two sample means, based on the amount of variability, or standard error, in the sample means. Formulas for the t-test can be found in statistics textbooks for two cases: equal variance and unequal variance between the two sets of samples. The assumption of unequal variance would seem to be more appropriate for gene expression analysis, especially if the active genes have greater variability in gene expression than inactive ones have.

In addition, there is a version of the t-test for paired samples. This version might be applicable to matched biopsy samples; that is, when normal and tumour tissues are obtained from the same patient. Such samples might be obtained from the centre and margins of a surgical tumour resection. The advantage of matched samples is that they remove variability between patients or animals and thereby make comparisons more sensitive.

The t-test assumes that the replicate data have an underlying normal distribution. This assumption is somewhat reasonable, especially if the replicate samples are relatively homogeneous. Note that the assumption of normality here is different from the assumption of normality that we discussed previously in the two-sample case. In that discussion, we considered the distribution of relative expression values over heterogeneous genes in a given sample, not for a given gene over homogeneous replicate samples. In most cases, we have relatively few replicate samples and it is difficult to test for normality in only a few data points [39]. Therefore, we often adopt the assumption of normality because it is hard to prove otherwise.

If the assumption of normality does hold, the t statistic can be compared with the appropriate t distribution to determine a p value. However, we must be careful about assigning p values to data that have undergone normalization or bias correction. These procedures attempt to make the expression levels approximately the same across samples, thereby artificially reducing the amount of variability. In turn, lower values for variability lead to higher t values and a large number of false-positive results.

It is not yet clear how best to assign p values after the data have been normalized, or when the normal assumption does not hold. One possibility is to apply a resampling method based on a permutation test [45]. A permutation test creates bootstrap samples by reassigning category labels randomly. For example, suppose the data are derived from four tumour and four normal samples. In each bootstrap sample, for each gene, we create a permutation of four tumour labels and four normal labels and assign those labels to the values before computing the t statistic. The distribution of extreme t statistics indicates the appropriate p value to assign. These types of analysis are currently under investigation by several researchers.

Non-parametric methods

The t-test is an example of a parametric approach, because it depends on certain parameters, such as the variances for the underlying normal distributions. We may also consider a non-parametric approach to the problem, where we do not assume that the data follow any particular type of distribution. In a non-parametric test, we replace the quantitative expression values with ranks or true–false assessments and use these new values to compute some statistic.

One standard non-parametric test that has been used to analyse microarray data is the Mann–Whitney test. In this test, we group together the values from the two samples and calculate the sum of the ranks that come from each sample. If this rank sum statistic is smaller or larger than we would expect under the null hypothesis, then the samples are statistically different from each other for this gene. Instead of using ranks, we can compute the rank sum statistic by using

true–false comparisons of the data instead; specifically, by evaluating all pairs of values from sample 1 and from sample 2.

The ranking and pairwise formulation are mathematically equivalent. However, the pairwise formulation is particularly appropriate for Affymetrix arrays. The Affymetrix software uses a proprietary algorithm to judge pairs of probe sets, resulting in a qualitative difference call that takes one of five possible values: increased, marginally increased, no change, marginally decreased, or decreased. We can use these qualitative calls in the pairwise formulation of the Mann–Whitney test to compute the rank sum statistic.

At our institution, we have used the Mann–Whitney method to identify overexpressed genes. In an experiment to determine the effect of captopril on cardiac gene expression, my colleagues studied heart tissue from rats in which myocardial infarction (MI) was induced surgically [47]. Six samples were from rats treated with captopril and six were from untreated rats. In addition, there were also six samples from control rats which received only a sham operation. A pairwise comparison of the MI samples with the sham samples using the Mann–Whitney method identified 37 genes that were significantly induced and six that were significantly repressed. Another pairwise comparison of the captopril–treated MI samples with the untreated MI samples showed that ten of the 37 genes had reversal of their changes in gene expression.

The changes in gene expression identified by the Mann–Whitney method were confirmed by performing quantitative PCR using a TaqMan sequence detector, showing that the Mann–Whitney method yields few false-positives. However, because difference calls are qualitative and somewhat conservative, the Mann–Whitney method appears to be relatively insensitive for identifying true changes in gene expression.

Discovering condition subtypes

In performing replicate samples, we may be interested not only in reducing variability, but also in studying it. We may wish to know, for example, whether we can discover subtypes among the replicate samples. Although our replicate samples were chosen originally to be similar, they may in fact be heterogeneous, consisting of two or more subtypes. Discovering previously unknown subtypes from the data is an example of unsupervised pattern recognition, for which cluster analysis is the prototypical method. Cluster analysis has been applied extensively to microarray data [30], usually to identify subgroups of genes, rather than samples. However, recent studies have started to use cluster analysis to identify subgroups of samples. This type of cluster analysis is useful in identifying candidate genes, because some genes are expressed only in particular tissue subtypes. Knowing these subtypes allows us to refine our search for genes of interest.

There are several methods for performing cluster analysis and many have already been applied to microarray data for clustering genes, including hierarchical clustering [16,48,49], k-means clustering [50–52], and self-organizing maps [53,54]. In addition, new types of cluster analysis techniques are being developed specifically for microarray data [55,56]. Cluster analysis methods differ along several attributes [57]. They can be either hierarchical or partitional, depending on the type of structure that they impose on the data. A hierarchical classification organizes the data into a dendrogram or tree structure, whereas a partitional method organizes the data into a single collection of groups. A hierarchical clustering specifies a sequence of nested partitions and we can obtain a single partition by cutting the dendrogram at a particular level.

Clustering algorithms can also be distinguished by whether they operate in an agglomerative or a divisive fashion. An agglomerative algorithm starts with each individual data element in its own cluster and then combines them to form larger clusters. In contrast, a divisive algorithm starts with the entire set of data in a single cluster and then subdivides the cluster to form smaller clusters. In order to perform a clustering analysis, we need to assess the similarity of two samples. Specifically, we require some function that takes two expression signatures (as defined in Figure 1) and produces some distance measure. The goal of cluster analysis is to produce clusters where this distance measure is small within clusters and large between clusters.

One example of sample clustering is a recent analysis of adult lymphoid malignancy [58]. In this analysis, researchers studied 96 samples of normal and malignant lymphocytes, including samples from patients with diffuse large B-cell lymphoma (DLBCL), follicular lymphoma (FL), and chronic lymphocytic leukaemia (CLL). Although these lymphomas are known to be distinct clinically, they were considered to be a single set of replicate samples for cluster analysis.

Hierarchical clustering of the data showed that FL and CLL samples were relatively similar to normal B-cells. However, DLBCLs had higher expression of several genes, especially those involved in cellular proliferation. Lower levels of the dendrogram revealed the presence of two distinct subtypes of DLBCLs, according to their expression signatures. One subtype had an expression signature similar to germinal centre B-cells and the other resembled activated peripheral blood B-cells. Interestingly, these subtypes of DLBCLs appeared to correlate with clinical outcome, with patients with germinal centre-like DLBCL having better survival rates than those with activated B-cell-like DLBCL.

Analysis of multiple conditions

Until now, we have considered at most two conditions, such as tumour versus normal. However, in any given

undertaking, we are likely to accumulate microarray data for multiple conditions. For example, my colleagues and I have applied microarrays to a variety of tumours, including those from lung, breast, and prostate tissues. Although microarray experiments involving multiple conditions have been performed [16], the corresponding analyses have largely focused on cluster analyses of genes. However, we can also exploit data from multiple conditions to help identify candidate genes. With multiple tissue types, for instance, we can identify genes that are not only tumour-specific, but also tissue-specific. Analysing data from multiple conditions requires more sophisticated types of statistical tools, which we discuss here.

Analysis of variance

The *t*-test discussed previously is designed to compare two sets of replicates. For multiple sets of replicate samples, we can perform an *F*-test. Like the *t*-test, the *F*-test assumes that the replicates under each condition are generated from a normal distribution, each with some mean $\mu+\alpha_i$, where μ is the mean over all conditions and α_i is the mean of condition i relative to the overall mean.

To perform the *F*-test, we essentially compare two estimates of variance. One estimate of variance comes from the variability of expression levels within groups. The other estimate comes from the variability of mean expression levels between groups. If the between-group estimate is much higher than the within-group estimate, we have evidence that the groups do not share the same mean. Hence, the *F*-test is also known as a one-way analysis of variance.

Note that the *F*-test will yield a positive result whenever any of the conditions is statistically different from the other conditions. Therefore, after the *F*-test yields gene candidates, we need to examine the expression profile for each gene further to determine which of the conditions is in fact significantly different from the others. Also, as with the *t*-test, we must consider the issue of multiple hypothesis testing and incorporate appropriate procedures to compensate for the large volume of genes being tested.

Pseudoprofiles

The *F*-test is useful for identifying genes that are overexpressed or underexpressed in any one of several conditions. However, we often have a particular expression profile in mind. For instance, we may wish to identify genes that are expressed highly in lung tumours, but at low levels in normal lung tissue and in other types of tumours. One way to identify genes with a particular behavior is to use the desired expression profile as a pseudoprofile and to find genes that match the pseudoprofile.

For most pseudoprofile analyses, we are interested in relative expression values, rather than absolute values, across the multiple conditions. Therefore, we must standardize each expression profile to reflect relative changes in expression. One way to do this is to subtract the mean value $\overline{X}$ for each profile $\langle X_1, X_2, \ldots, X_N\rangle$. Another method would be to subtract a particular value, such as X_1 from each profile. That value would essentially make each profile start at a expression level of zero and would be a natural choice for time series experiments.

In order to find genes that match a given pseudoprofile, we need some way to measure the similarity of each gene's profile with the pseudoprofile. There are a variety of methods for making this measurement. Many of these metrics are useful for performing cluster analysis, where we also need to know the similarity between pairs of genes.

One method is to compute the Euclidean distance. Suppose a profile has (standardized) values $\langle X_1, X_2, \ldots, X_N\rangle$ and the pseudoprofile has (standardized) values $\langle Y_1, Y_2, \ldots, Y_N\rangle$. Then the Euclidean distance is

$$\|X-Y\| = \sqrt{\sum_{i=1}^{N}(X_i-Y_i)^2} \qquad (1)$$

Another way to compute similarity between two profiles is to use their linear correlation coefficient, also known as Pearson's *r* coefficient. The linear correlation coefficient measures the product of the deviation of a profile from its mean and the deviation of the pseudoprofile from its mean, and then normalizes that product:

$$r = \frac{\sum_{i=1}^{N}(X_i-\overline{X})(Y_i-\overline{Y})}{\sqrt{\sum_{i=1}^{N}(X_i-\overline{X})^2}\sqrt{\sum_{i=1}^{N}(Y_i-\overline{Y})^2}} \qquad (2)$$

Because of the normalization, the linear correlation coefficient does not change when the profile or pseudoprofile is subjected to linear transformations, such as scaling factors.

One difficulty with pseudoprofiles is that they are quite specific; we must choose a particular expression value for each point in the profile. However, we may want to specify only vaguely that the expression in some conditions should be higher than in other conditions. To make this type of specification, we can use similarity measures based on ranks, rather than precise numerical quantities. Several types of rank correlation measures exist; one well-known measure is Spearman's rank-order correlation coefficient. For this measure, we replace each value X_i by its rank among all X_i, and each value Y_i by its rank among all Y_i. Let the resulting ranks be R_i and S_i, respectively. Then the rank-order correlation coefficient is

$$r_s = \frac{\sum_{i=1}^{N}(R_i-\overline{R})(S_i-\overline{S})}{\sqrt{\sum_{i=1}^{N}(R_i-\overline{R})^2}\sqrt{\sum_{i=1}^{N}(S_i-\overline{S})^2}} \qquad (3)$$

A pseudoprofile effectively prioritizes the genes according to their distance from the pseudoprofile. We can visualize the ranking through a graphical interface. For example, in Figure 4, we show how the pseudoprofile method selects genes that are expressed highly in colon tumours, but are expressed at average levels in normal lung tissue as well as breast and lung tumours. It remains a research issue whether we can assign appropriate p values to pseudoprofile matches, especially if the distributional assumptions are not satisfied.

Supervised pattern recognition

Pseudoprofiles rank genes according to their similarity to a given expression profile. If we specify a particular distance cut-off, we are essentially categorizing genes into two groups: those that are similar and those that are dissimilar to the given profile. In essence, we are 'training' the computer by giving it a certain expression profile, and asking it to categorize genes according to the training rule.

This training approach to gene identification in the most general case is called machine learning [59]. In the machine learning paradigm, the training rule is usually not given explicitly. Instead, we train the computer by providing a set of examples or cases where we know the category. The computer then processes these training data to discover the best discrimination rule for categorizing future data.

Machine learning is often called supervised pattern recognition, because we provide the computer with an initial set of categorized data. Supervised pattern recognition contrasts with unsupervised pattern recognition, where we ask the computer to discover categories without any prior training cases. Cluster analysis is the standard procedure for performing unsupervised pattern recognition.

For the purpose of identifying candidate genes, one could imagine that we have some gene families of particular interest and we want to identify additional members of each family based on their gene expression. If we use the existing gene families as a training set, we could apply the machine-learning paradigm to identify genes of interest.

There are various methods for performing machine learning, each differing in the type of discrimination rule that the computer tries to discover [60]. For example, in tree-based machine learning, the discrimination rule is represented by a decision tree, where a series of yes–no questions determines the final category. In neural network-based machine learning, the discrimination rule is represented by a neural network. and the computer must determine the appropriate

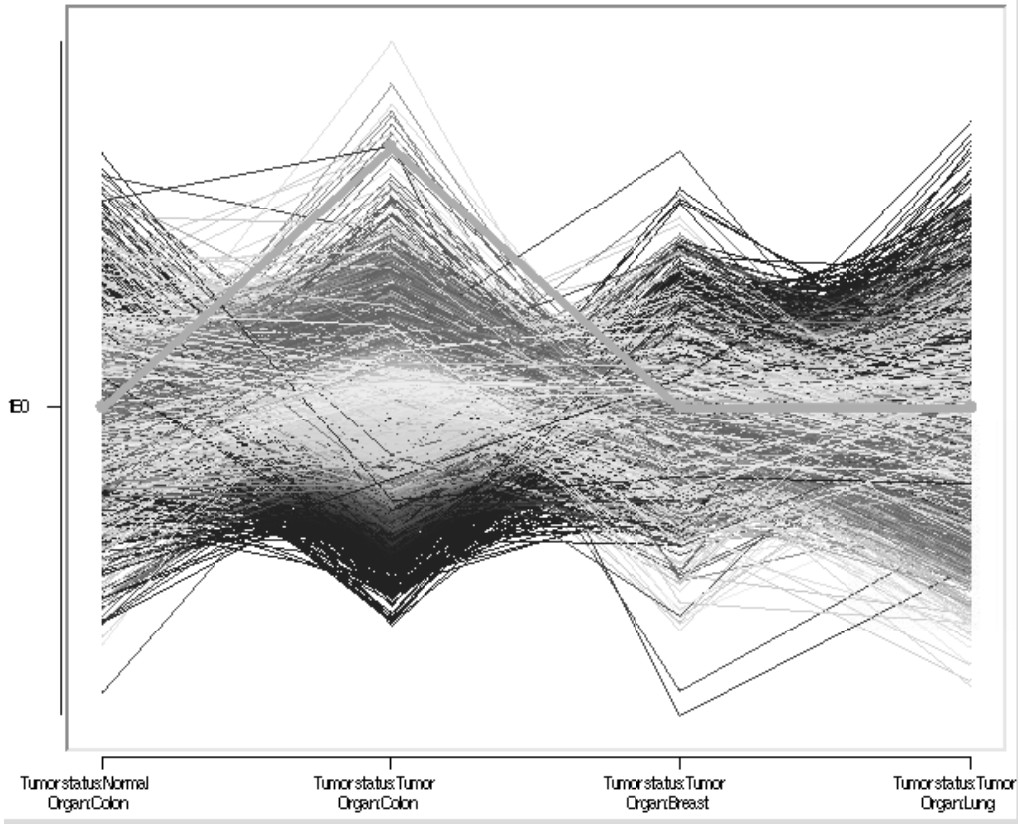

Figure 4. Gene expression profiles and a pseudoprofile. This figure shows gene expression profiles for cDNA microarrays covering four conditions. From left to right, these conditions are normal colon tissue, colon tumour, breast tumour, and lung tumour. A pseudoprofile is shown as a thick line, intended to identify genes with high expression in colon tumours relative to the other conditions. Each profile is coloured to indicate its similarity to the pseudoprofile, with red indicating high similarity and blue representing high dissimilarity. The rendition shown here, though, depicts these colors in grey scale

structure and parameters of the network needed to categorize the training data. In discriminant-based machine learning, the discrimination rule is represented by a formula and the values of the formula are used to classify data.

The supervised pattern recognition approach for microarray data is still new and researchers are still exploring various techniques. One recent study [61] uses a type of discriminant-based machine learning called support vector machines to categorize 2467 yeast genes into one of six functional classes, based on data from 79 different microarray experiments. The authors found that support vector machines outperformed four other types of machine learning methods.

Supervised pattern recognition can be applied not only to genes, but also to samples. One example of this type of analysis involved 38 samples of bone marrow, of which 27 came from patients with acute lymphoblastic leukaemia (ALL) and 11 came from those with acute myeloid leukaemia (AML) [62]. Researchers essentially used a *t*-type analysis to identify 1100 genes that were different in the two conditions. They then used the 50 genes with the greatest difference to construct a 'voting'-based discrimination rule, similar to a method called nearest neighbours. Their predictor was able to classify new samples accurately as either AML or ALL.

Covariate analysis

Each gene and each sample that we study with a microarray may be associated with one or more covariates. A covariate is a variable that contains contextual information for a sample or gene. One example of contextual information, which we have already covered extensively, is the tissue type. Replicate samples are simply a set of samples that share the same value for the tissue type covariate. Tissue type is a categorical or factor covariate, one that can assume one of several discrete values. Other types of covariates may be continuous. For example, we might quantify the percentage of tumour cells and normal cells in each sample and associate the resulting number with each sample as a covariate.

We have already used covariates implicitly to form groups of replicate samples. But if we represent covariates explicitly, we can perform additional types of statistical analyses. Covariate analysis has not yet been applied widely in the microarray literature. However, we anticipate that it will become increasingly important and we suggest some avenues for further research.

Regression analysis

One well-studied approach to handling covariate data is regression analysis, which is a type of statistical method called modelling [63]. In modelling, we attempt to find some formula or procedure that replaces our observed data with a set of fitted data. The basic goal is to find a model that is as simple as possible, while also producing a close fit to the observed data. The resulting model should provide insight about the underlying processes and parameters that generated the data.

In regression modelling, we try to fit the data with a formula that consists of a sum of predictor effects, with each predictor coming from a covariate or combination of covariates. The simplest type of regression model is a linear regression model, in which the effects are represented by a coefficient multiplied by the predictor [64]. For example, suppose that we have a tumour percentage X_i for each microarray i. For a given gene j, we might predict its expression levels Y_{ji} over all microarrays as with the following linear regression model:

$$Y_{ji} = \mu_j + \beta_j X_i + \varepsilon_{ji} \quad (4)$$

In this equation, μ_j is the expression level for the gene in normal tissues and β_j describes how strongly the expression level increases in tumours. The term ε_{ji} is an error term that is normally distributed with mean zero.

When we fit this model to the observed data for gene j, we obtain an estimate of the coefficient β_j, as well as the statistical significance of this estimate. A coefficient β_j that equals zero indicates that the expression of gene j does not depend linearly on the tumour percentage. In contrast, a positive or negative coefficient indicates that the expression of gene j increases or decreases linearly with the tumour percentage. The p value indicates the degree to which β_j is different from zero, based on the standard error of the estimate. Therefore, using covariate analysis, we can identify candidate genes as those that have coefficients that are statistically significant and are strongly positive or negative.

We can apply regression models that are progressively more complex. For instance, whereas a linear regression model has only linear effects, an additive model permits non-linear effects; the model-fitting process tries to estimate the shape of the non-linear functions [63]. Such a model might be useful for fitting microarray data from time series experiments. There also exist other types of models, including discriminant models, decision trees, and neural networks, and these types of data analysis may become useful tools as our microarray data increase in size and complexity.

Correlation analysis

In the previous section, we discussed covariates that are essentially independent variables. They are controlled by the experimenter, or can be described directly from the sample. However, we can consider covariate information that are dependent variables, perhaps derived from some other experiment or test. We can then ask whether the gene expression data and covariate data are correlated.

One example of a large covariate study involves a linkage between gene expression data and molecular pharmacology data [65]. The molecular pharmacology data were derived from an existing study of 60 human cancer cell lines from the National Cancer Institute. Each cell line has been tested with over 70 000 drug compounds in order to determine the cell's sensitivity to various drugs. Drug activities have been quantified for each combination of cell line and drug compound. The authors of this study also applied each cell line to a cDNA microarray in order to obtain gene expression data.

Therefore, the combined data could be represented as two matrices. One matrix contained drug activity levels, organized by samples versus drug compounds. The other matrix was the gene expression data, organized by samples versus genes. In the two matrices, expression or drug activity profiles spanned the same cell lines and could be compared with one another. Accordingly, the researchers in the study computed correlation coefficients between various pairs of drug activity and gene expression profiles and found several instances of correlation between gene expression and drug activity.

Many of the significant correlations appeared to be negative. For example, expression of dihydropyrimidine dehydrogenase (DPYD) was negatively correlated with sensitivity to 5-fluorouracil (5-FU). This makes physiological sense because DPYD is the rate-limiting enzyme in 5-FU catabolism. Thus, cells with high levels of DPYD have lower levels of the active phosphorylated forms of 5-FU.

Discussion

As we have seen, microarray data can be interpreted in many ways. The conclusions that can be drawn from a given set of data depend heavily on the particular choice of data analysis. Much of the data analysis depends on such low-level considerations as normalization and on such basic assumptions as normality. For this reason, we must be careful about inferences made from microarray data.

The success of a microarray experiment depends also on the quality of the samples being assayed. Because of variability across different samples, it will become increasingly important to perform microarray experiments on replicate samples. These samples need to be chosen carefully to represent the population being studied. Moreover, biopsy samples themselves consist of multiple cell types, which also add to variability in microarray experiments. Careful preparation of tissue specimens and recent techniques such as laser capture microdissection [66] may help to reduce this source of variability.

Covariate information will also become increasingly important in microarray experiments, even though such data have not yet been used widely in microarray data analysis. We have shown how we might exploit information about the tumour percentage in a given sample to increase our ability to identify candidate genes. Other types of covariate information will depend on the accurate pathological description of biological specimens.

Microarray experimentation and analysis are still in their infancy and we lack widely agreed-upon methods for storing and analysing data and communicating results. In contrast, for sequence data, we have succeeded in developing standard databases, such as GenBank, and standard tools, such as BLAST [67]. Calls have been made to create centralized databases of microarray data, including the recent Gene Expression Onmibus proposal from the National Center for Biotechnology Information and a similar proposal from the European Bioinformatics Institute. However, microarray data are much more complex than sequence data because microarray results depend heavily on particular biological conditions [68]. Standardization of microarray data will therefore depend on a general ontology for describing environmental and experimental conditions, as well as a standard nomenclature for cell types and tissue specimens. Ironically, these standardization efforts will probably require revision as more microarray experiments are performed. As microarray experiments move out of the realm of molecular and cellular biology and into the field of clinical medicine, they are likely to redefine our existing definitions of cell, tissue, and disease categories, and allow us to understand pathophysiology with renewed precision.

Acknowledgements

I would like to thank William Forrest, Mary Gerritsen, Steve Guerrero, Michael Ostland, Nicholas Paoni, Victoria Smith, Jerry Tang, Michael Ward, Mickey Williams, and William Wood for stimulating discussions about microarray experimental design and data analysis.

References

1. Collins FS, Patrinos A, Jordan E, Chakravarti A, Gesteland R, Walters L. New goals for the U.S. human genome project: 1998–2003. *Science* 1998; **282**: 682–689.
2. Lockhart DJ, Winzeler EA. Genomics, gene expression and DNA arrays. *Nature* 2000; **405**: 827–836.
3. Bowtell DDL. Options available – from start to finish – for obtaining expression data by microarray. *Nature Genet* 1999; **21**: 25–32.
4. Chen Y, Dougherty ER, Bittner ML. Ratio-based decisions and the quantitative analysis of cDNA microarray images. *J Biomed Opt* 1997; **2**: 364–374.
5. Cheung VG, Morley M, Aguilar F, Massimi A, Kucherlapati R, Childs G. Making and reading microarrays. *Nature Genet* 1999; **21**: 15–19.
6. Ermolaeva O, Rastogi M, Pruitt KD, *et al*. Data management and analysis for gene expression arrays. *Nature Genet* 1998; **20**: 19–23.
7. Aach J, Rindone W, Church GM. Systematic management and analysis of yeast expression data. *Genome Res* 2000; **10**: 431–445.

8. Ringwald M, Eppig JT, Kadin JA, Richardson JE. GXD: a gene expression database for the laboratory mouse: current status and recent enhancements. *Nucleic Acids Res* 2000; **28**: 115–119.
9. Miller G, Fuchs R, Lai E. IMAGE cDNA clones, UniGene clustering, and ACeDB: an integrated resource for expressed sequence information. *Genome Res* 1997; **7**: 1027–1032.
10. Bassett DE Jr, Eisen MB, Boguski MS. Gene expression informatics – it's all in your mine. *Nature Genet* 1999; **21**: 51–55.
11. Kanehisa M. *Post-Genome Informatics*. Oxford University Press: Oxford, 2000.
12. Somogyi R, Sniegoski CA. Modeling the complexity of genetic networks: understanding multigenic and pleiotropic regulation. *Complexity* 1996; **1**: 45–63.
13. Kanehisa M, Goto S. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res* 2000; **28**: 29–34.
14. Ashburner M, Ball CA, Blake JA, *et al*. Gene ontology: tool for the unification of biology. *Nature Genet* 2000; **25**: 25–29.
15. DeRisi JL, Iyer VR, Brown PO. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science* 1997; **278**: 680–686.
16. Eisen MB, Spellman PT, Brown PO, Botstein D. Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A* 1998; **95**: 14863–14868.
17. Roth FP, Hughes JD, Estep PW, Church GM. Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *Nature Biotechnol* 1998; **16**: 939–945.
18. van Helden J, Andre B, Collado-Vides J. Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *J Mol Biol* 1998; **281**: 827–842.
19. Brazma A, Jonassen I, Vilo J, Ukkonen E. Predicting gene regulatory elements in silico on a genomic scale. *Genome Res* 1998; **8**: 1202–1215.
20. Zhang MQ. Large-scale gene expression data analysis: a new challenge to computational biologists. *Genome Res* 1999; **9**: 681–688.
21. Debouck C, Goodfellow PN. DNA microarrays in drug discovery and development. *Nature Genet* 1999; **21**: 48–50.
22. Gray NS, Wodicka L, Thunnissen AMWH, *et al*. Exploiting chemical libraries, structure, and genomics in the search for kinase inhibitors. *Science* 1998; **281**: 533–538.
23. Marton MJ, DeRisi JL, Bennett HA, *et al*. Drug target validation and identification of secondary drug target effects using DNA microarrays. *Nature Med* 1998; **4**: 1293–1301.
24. Perou CM, Jeffrey SS, van de Rijn M, *et al*. Distinctive gene expression patterns in human mammary epithelial cells and breast cancers. *Proc Natl Acad Sci U S A* 1999; **96**: 9212–9217.
25. Perou CM, Sorlie T, Eisen MB, *et al*. Molecular portraits of breast tumours. *Nature* 2000; **406**: 747–752.
26. Alon U, Barkai N, Notterman DA, *et al*. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc Natl Acad Sci U S A* 1999; **96**: 6745–6750.
27. Heller RA, Schena M, Chai A, *et al*. Discovery and analysis of inflammatory disease-related genes using cDNA microarrays. *Proc Natl Acad Sci U S A* 1997; **94**: 2150–2155.
28. McCaffrey TA, Fu C, Du B, *et al*. High-level expression of Egr-1 and Egr-1-inducible genes in mouse and human atherosclerosis. *J Clin Invest* 2000; **105**: 653–662.
29. Kaminski N, Allard JD, Pittet JF, *et al*. Global analysis of gene expression in pulmonary fibrosis reveals distinct programs regulating lung inflammation and fibrosis. *Proc Natl Acad Sci U S A* 2000; **97**: 1778–1783.
30. Brazma A, Vilo J. Gene expression data analysis. *FEBS Lett* 2000; **480**: 17–24.
31. Fodor S, Rava R, Huang X, Pease A, Holmes C, Adams C. Multiplexed biochemical assays with biological chips. *Nature* 1993; **364**: 555–556.
32. Lipshutz RJ, Fodor SPA, Gingeras TR, Lockhart DJ. High density synthetic oligonucleotide arrays. *Nature Genet* 1999; **21**: 20–24.
33. Schena M, Shalon D, Davis RW, Brown PO. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* 1995; **270**: 467–470.
34. Duggan DJ, Bittner M, Chen Y, Meltzer P, Trent JM. Expression profiling using cDNA microarrays. *Nature Genet* 1999; **21**: 10–14.
35. Mardia KV, Kent JT, Bibby JM. *Multivariate Analysis*. Academic Press: London, 1979.
36. Schena M, Shalon D, Heller R, Chai A, Brown PO, Davis RW. Parallel human genome analysis: microarray-based expression monitoring of 1000 genes. *Proc Natl Acad Sci U S A* 1996; **93**: 10614–10619.
37. McLachlan GJ, Krishnan T. *The EM Algorithm and Extensions*. Wiley: New York, 1997.
38. DeRisi J, Penland L, Brown PO, *et al*. Use of a cDNA microarray to analyse gene expression patterns in human cancer. *Nature Genet* 1996; **14**: 457–460.
39. Rice JA. *Mathematical Statistics and Data Analysis* (2nd edn). Duxbury Press: Belmont, CA, 1995.
40. Efron B, Tibshirani R. *An Introduction to the Bootstrap*. Chapman & Hall: London, 1993.
41. McLachlan G, Peel D. *Finite Mixture Models*. Wiley: New York, 2000.
42. Lee MLT, Kuo FC, Whitmore GA, Sklar J. Importance of replication in microarray gene expression studies: statistical methods and evidence from repetitive cDNA hybridizations. *Proc Natl Acad Sci U S A* 2000; **97**: 9834–9839.
43. Shaffer JP. Multiple hypothesis testing. *Annu Rev Psychol* 1995; **46**: 561–584.
44. Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J R Stat Soc B* 1995; **57**: 289–300.
45. Westfall PH, Young SS. *Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment*. Wiley: New York, 1993.
46. Bartosiewicz M, Trounstine M, Barker D, Johnston R, Buckpitt A. Development of a toxicological gene array and quantitative assessment of this technology. *Arch Biochem Biophys* 2000; **376**: 66–73.
47. Jin H, Yang R, Awad TA, *et al*. Effects of early ACE inhibition on cardiac gene expression following acute myocardial infarction. *Circulation* 2001; **195**: 736–742.
48. Chu S, DeRisi J, Eisen M, *et al*. The transcriptional program of sporulation in budding yeast. *Science* 1998; **282**: 699–705.
49. Eickhoff H, Schuchhardt J, Ivanov I, *et al*. Tissue gene expression analysis using arrayed normalized cDNA libraries. *Genome Res* 2000; **10**: 1230–1240.
50. Hartigan JA, Wong MA. A K-means clustering algorithm. *Appl Stat* 1979; **28**: 100–108.
51. Tavazoie S, Hughes JD, Campbell MJ, Cho RJ, Church GM. Systematic determination of genetic network architecture. *Nature Genet* 1999; **22**: 281–285.
52. Herwig R, Poustka AJ, Müller C, Bull C, Lehrach H, O'Brien J. Large-scale clustering of cDNA fingerprinting data. *Genome Res* 1999; **9**: 1093–1105.
53. Kohonen T. *Self-Organizing Maps*. Springer: New York, 1997.
54. 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 U S A* 1999; **96**: 2907–2912.
55. Ben-Dor A, Shamir R, Yakhini Z. Clustering gene expression patterns. *J Comput Biol* 1999; **6**: 281–297.
56. Heyer LJ, Kruglyak S, Yooseph S. Exploring expression data: identification and analysis of coexpressed genes. *Genome Res* 1999; **9**: 1106–1115.
57. Jain AK, Dubes RC. *Algorithms for Clustering Data*. Prentice Hall: Englewood Cliffs, NJ, 1988.
58. Alizadeh AA, Eisen MB, Davis RE, *et al*. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature* 2000; **403**: 503–511.
59. Ripley BD. *Pattern Recognition and Neural Networks*. Cambridge University Press: Cambridge, 1996.

60. Venables WN, Ripley BD. *Modern Applied Statistics with S-Plus* (2nd edn). Springer: New York, 1994.
61. Brown MPS, Grundy WN, Lin D, *et al*. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc Natl Acad Sci U S A* 2000; **97**: 262–267.
62. Golub TR, Slonim DK, Tamayo P, *et al*. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* 1999; **286**: 531–537.
63. Hastie TJ, Tibshirani RJ. *Generalized Additive Models*. Chapman & Hall: London, 1990.
64. Weisberg S. *Applied Linear Regression* (2nd edn). Wiley: New York, 1985.
65. Scherf U, Ross DT, Waltham M, *et al*. gene expression database for the molecular pharmacology of cancer. *Nature Genet* 2000; **24**: 236–244.
66. Bonner RF, Emmert-Buck M, Cole K, *et al*. Laser capture microdissection: molecular analysis of tissue. *Science* 1997; **278**: 1481–1483.
67. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. *J Mol Biol* 1990; **215**: 403–410.
68. Brazma A, Robinson A, Cameron G, Ashburner M. One-stop shop for microarray data. *Nature* 2000; **403**: 699–700.

