Cellular Oncology 26 (2004) 279–290
IOS Press

Reviews

Microarray data analysis: From hypotheses to conclusions using gene expression data

Nicola J. Armstrong [a] and Mark A. van de Wiel [b,*]
[a] *Department of Mathematics, Vrije Universiteit, Amsterdam, The Netherlands*
[b] *Microarray Facility, Vrije Universiteit, Amsterdam, The Netherlands*

Abstract. We review several commonly used methods for the design and analysis of microarray data. To begin with, some experimental design issues are addressed. Several approaches for pre-processing the data (filtering and normalization) before the statistical analysis stage are then discussed. A common first step in this type of analysis is gene selection based on statistical testing. Two approaches, permutation and model-based methods are explained and we emphasize the need to correct for multiple testing. Moreover, powerful approaches based on gene sets are mentioned. Clustering of either genes or samples is frequently performed when analyzing microarray data. We summarize the basics of both supervised and unsupervised clustering (classification). The latter may be of use for creating diagnostic arrays, for example. Construction of biological networks, such as pathways, is a statistically challenging but complex task that is a relatively new development and hence mentioned only briefly. We finish with some remarks on literature and software. The emphasis in this paper is on the philosophy behind several statistical issues and on a critical interpretation of microarray related analysis methods.

1. Introduction

In recent years biology has greatly benefited from the development of microarray technology which allows the simultaneous measurement of expression levels in thousands of genes in a biological sample. First produced in the Brown lab at Stanford University [31], many laboratories worldwide are now making their own arrays, in addition to the availability of commercial vendors such as Affymetrix (Santa Clara, CA) and Agilent (Palo Alto, CA).

A microarray is a glass slide containing anywhere between 100 to 10,000 or more tiny spots consisting of what are known as probe sequences. Depending on the platform used, probes are either single-stranded cDNA, long oligonucleotides (60–70 bp) or short oligonucleotides (25 bp, Affymetrix). Target RNA is generally extracted from samples of interest (e.g. cancer tumors or cell lines), reverse transcribed into cDNA, labeled with fluorescent dye and then hybridized to the array. Most common are the so-called two color arrays, where two different samples are labeled with different dyes (Cy3, green and Cy5, red) and then hybridized simultaneously to the same slide.

The main idea behind this technique is that the fluorescent intensity of a spot is equivalent to the amount of RNA expressed in the sample. In this way, biologists can begin to identify genes involved in specific processes or diseases by looking, for example, at differences between cell lines, cancer types or response to drug treatment. Predictions can also be made regarding gene function – if an unknown gene has a similar expression pattern to a well-known group of genes, then perhaps the unknown gene has a similar function. Likewise, by looking at gene knockouts or RNAi experiments, genetic pathways might become clearer. These are just a few of the questions biologists can seek to answer with the aid of microarrays. The design and analysis of such experiments plays a crucial role in whether the answers can be elucidated from the data collected. In this article we aim to provide a brief introduction to the statistical methods that are being used to analyze microarray experiments. The main issues of design, pre-processing, determination of differential

*Corresponding author: M.A. van de Wiel. Present address: Department of Mathematics and Computer Science, Technische Universiteit Eindhoven, P.O. Box 513, 5600 MB Eindhoven, The Netherlands. Fax: +31 40 2465995; E-mail: m.a.v.d.wiel@tue.nl.

expression and clustering/classification are presented as well as recent attempts to build regulatory networks. Finally, we give an overview of helpful literature and software for this area.

2. Design

As with any experiment, the design will ultimately dictate whether the questions deemed important by the biologist can, in the end, be answered. We briefly describe some of the important design issues to consider, and refer the reader to two informative overview papers for more detail [7,43]. The final design used for a microarray experiment will be constrained by both the type of arrays used and the number available as well as by biological constraints, such as RNA availability. It is also important to keep in mind that the software that will be used to analyze the data should be able to cope with the chosen design (at this time, Resolver (Rosetta BioSoftware, Seattle, WA) and MAS 5.0 (Affymetrix) are unable to analyze factorial experiments or so-called loop designs).

Generally, biologists are interested in more than one question which they hope to answer with a single microarray experiment. Different designs may answer different questions optimally so that the biologist should prioritize which questions are most important. Consider the case of a time course experiment where samples are extracted at four different time points: T_1, T_2, T_3 and T_4. The biologist might be interested in comparing gene expression between T_1 and all other time points or between consecutive times points T_1–T_2, T_2–T_3 and T_3–T_4 or both. The design which is optimal for answering the former question might not provide the most accurate answers to the latter and vice versa (Fig. 1A–C). Hence the biologist may have to choose which differences are of most interest in a particular experiment.

With Affymetrix arrays and other one-color platforms there is no design issue concerning which samples to hybridize to each array. An array must be used for at least one (preferably more) representative sample for each different category. For two color arrays, this is a very real issue that needs to be addressed before the hybridizations are conducted. Two samples can be directly compared *in silico* very easily using the one color system, but in contrast, using two color arrays, they can only be directly compared if they are hybridized to the same array (if log ratios rather than intensity levels are used for analysis). Direct comparison

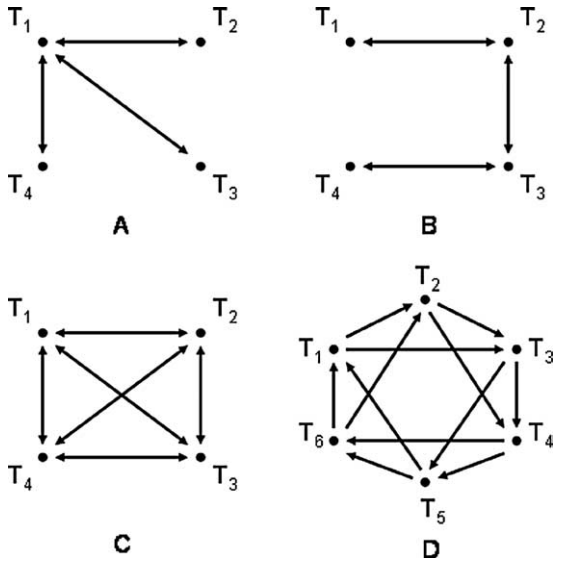

Fig. 1. (A) Optimal design, using 6 arrays, for comparing all time points to T_1. This design is also a common reference design, with T_1 being the reference. (B) Optimal design, using 6 arrays, for comparing gene expression between consecutive time points. (C) Optimal design, using 6 arrays, for comparing both consecutive time points and all time points to T_1. (D) Example of a loop design using 12 arrays to compare 6 time points. An edge or arrow connecting two time points indicates that these two samples are co-hybridized to the same array. The convention that the Cy5 labeled sample is at the head of the arrow and Cy3 at the tail is used. A double arrow indicates a dye swap.

of T_2 and T_3 will provide a more accurate picture of expression changes between the two time points than comparing both T_2 and T_3 to a common reference, i.e. an indirect design (Fig. 1A).

In some instances, such as determining differences in gene expression between different tumor types [1], a common reference design is the most suitable. This type of design is used in diagnostics and has the added advantage that if a suitable reference is chosen, the experiment can be readily extended to incorporate additional samples/patients as they become available. Comparisons between labs using the same reference may also be possible, or between experiments making it easier and desirable to build up databases of microarrays. However, if one wishes to detect differences between normal and tumor cells then the direct design approach is better. As the number of different conditions to be investigated increases, direct comparison designs rapidly increase in size, with large amounts of arrays being required. This means that they generally become unfeasible in terms of cost and, perhaps, with respect to the amount of RNA available. In these situations, more complicated designs such as the loop designs of Kerr

and Churchill [17] will most likely provide a suitable solution (Fig. 1D). It is also recommended to use dye swaps if they can easily be incorporated into the design in order to control for gene specific dye biases as well as the dye intensity differences [17].

Microarrays are an inherently noisy technology and as such replication is a good idea in order to reduce variability. Replicate spots on arrays are a good indicator of array quality, although they preferably should be printed in different regions of the array so that they are less dependent measurements. Each spot on the array does not necessarily correspond to a different gene. For example, several different probes for one gene might be spotted on an array. Differences in intensity levels among these probes may reflect technical differences between arrays and hence be a good indicator of quality or they may indicate that some probes themselves are of poor quality, for example, a probe sequence may not be unique to that particular gene. Technical replicates (i.e. use of target mRNA from the same extraction) of microarray slides will not remove biases present. Biological replicates (i.e. mRNA from different extractions, e.g. different mice) are more informative than technical replicates, although technical replicates can be useful for quality control, as outlined below. Whether the replicates come from the same or different sources depends on the experimental aims and restrictions. The issue of biological replication impacts the generalizability of the study, as does the issue of pooling RNA from more than one sample. If one wants to draw conclusions about an entire inbred strain of animals then it is better to use biological replicates of many random animals without pooling. However pooling may be necessary due to other constraints (e.g. amount of RNA available). At this stage there is little data or evidence available on the advantages or disadvantages of pooling and in many cases the decision is made based on other constraining factors.

3. Preprocessing

3.1. Image analysis

After the experiment has been designed and conducted, the slides are scanned and converted into images, generally 16 bit TIF files. Changing the scanner settings result in different images which can affect the experimental results. It is important that there is no saturation present (i.e. spots with the maximum possible intensity values) and that the linear range of the scanner is used. These images are then quantified using one of several available packages such as Imagene (BioDiscovery, El Segundo, CA) or GenePix (Molecular Devices, Union City, CA). For each spot on the array in the two color system, there are four quantities of interest: foreground and background intensities for each color. Different image analysis programs define the foreground and background areas of each spot according to different algorithms. The intensities are then generally measured as either the mean or median pixel value in the given region.

3.2. Quality assessment

A first crucial step after obtaining these data is to assess its quality. This usually starts with visual inspection of the images and plots of the raw data. An experienced eye will usually be able to judge whether any of the arrays in the set has inferior quality or whether some region(s) on the array(s) are unusual possibly due to scratches, printing tip effects or other spatial factors. Spatial plots of foreground, background or background subtracted intensity signals can also help identify regions of an array with too high (or low) signal, as can spatial plots of the log ratio values (Fig. 2). Both high and low signals should be randomly spread throughout the entire array. These types of plots can also be

Fig. 2. Spatial plot of M values after lowess normalization (M vs A plot for this data is shown in Fig. 3). Note the presence of spatial patterns. Data courtesy Prof. A.B. Smit, Department of Molecular and Cellular Neurobiology, Vrije Universiteit, Amsterdam, The Netherlands.

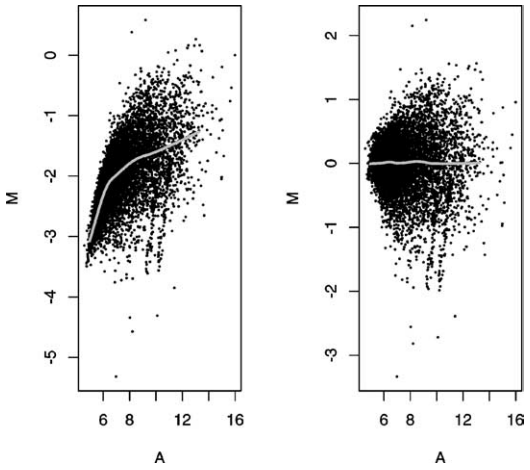

Fig. 3. M vs A plots for raw unadjusted data (on left) and after lowess normalization (right). The grey line seen in both pictures is the lowess line. The data are the same as in Fig. 2. Data courtesy Prof. A.B. Smit, Department of Molecular and Cellular Neurobiology, Vrije Universiteit, Amsterdam, The Netherlands.

made using quality control variables provided by the image analysis program, such as spot size or shape to further assess array quality. Another important kind of plot used frequently in microarray analysis is the so-called MA or RI plots (Fig. 3). These plot the log ratios, $\log_2 \frac{R}{G}$ or M values against the intensity or A values, $\frac{1}{2}\log_2(R \cdot G)$, where R and G represent the background adjusted intensity levels for a given spot. Most genes are not expected to be differentially expressed, hence the majority of points should lie in a cloud around $M = 0$.

A measure of the RNA-quality (e.g. from a NanoDrop spectrophotometer (Wilmington) or Nano LabChip (Agilent)), if available, may help determine whether an inferior array is due to bad biological material or a bad hybridization. Once such arrays have been discarded, the next step is: which spots on the arrays should be included in the analysis? This process is usually referred to as 'flagging'. Most image analysis software packages have their own specific flagging criteria, usually based on physical features of the spot (e.g. morphology) or on comparison with background values. For example, spots could be flagged if $\mathrm{FG} - \mathrm{BG} < 2\mathrm{sd(BG)}$, where FG and BG denote foreground and background intensity, respectively and sd(BG) denotes the standard deviation of background pixels. In two color arrays, often the entire spot is flagged when one of the two dye signals meets such a criterion, which may lead to loss of information. If the other dye gives a high signal then there is no indication that the probe is bad. In such cases, it may be more appropriate to set the low signal value to an upper value (such as 2sd(BG)) to obtain a conservative ratio estimate. Often technical replicates of some type are available. For example, the common reference design automatically results in replicates of the reference signal. When the number of technical replicates is small, all spots within a set of replicates may be flagged when they strongly disagree, e.g. [29], whereas in larger sets usually only outliers will be flagged. Formal flagging criteria based on repeatability are discussed in [16] for two color arrays and in [20] for Affymetrix arrays. When technical replicates are available, robust measures like trimmed means (which ignore outliers when computing the mean) and medians will be less sensitive to how one flags than the arithmetic mean.

Another useful indicator of array quality are control spots printed on the array. Negative control spots are DNA sequences that are known not to be present in the target samples, for instance plant or bacterial sequences when the targets are derived from mammalian cells. These spots should always be empty or contain no signal on the array. In contrast, positive control spots will always have high intensity in one channel due either to the sequence being from a housekeeping gene or the target being spiked with the complimentary sequence to ensure hybridization occurs.

3.3. Normalization

An important part of data preprocessing is normalization, which adjusts individual intensities so that comparisons can be made both within an array and between arrays in the experiment. Adjustments are necessary to remove differences which are purely technical and do not represent true biological variation. Examples of such differences are unequal RNA quantities, differences in labeling, systematic biases in measured expression levels, scanner settings, print-tip variation and sample plate origin. These differences, if left unadjusted, will hinder the ability to identify true differentially expressed genes and may increase the number of false positives found. In contrast to cDNA and long oligonucleotide arrays, the normalization of Affymetrix arrays is quite different. We neglect details here (see [5]) and concentrate on two color systems.

Within slide normalization is necessary in two color systems in order to adjust for the differences in intensity levels between the dyes. Red (Cy5) intensities are generally lower than green (Cy3) intensities, even in self–self hybridizations. However, even in one color systems it is advisable due to the possible existence

of spatial effects and the generally accepted observation that there is a systematic dependence on intensity levels. That is high intensity spots should be treated differently to low intensity spots. Older normalization techniques such as mean intensity methods or ANOVA models do not allow for this nonlinear phenomenon. Most normalization procedures assume that the majority of the genes present on the array are not differentially expressed so that the ratios should be 1. For special boutique arrays, that have only a few hundred spots, this assumption may not hold, and hence the normalization methods discussed here are not appropriate. The most commonly accepted form of adjustment is currently lowess (locally weighted least squares regression) or another form of nonlinear smoothing (Fig. 3). Although this method removes dye and intensity differences, it does not eliminate spatial patterns. If the arrays were printed using several printing tips and spatial patterns can be seen after ordinary lowess normalization, the lowess adjustment can be applied separately in each individual printing region. For some arrays, for example Agilent arrays, no print tips are used in the manufacturing process so there is no easy division of the array into sub grids in order to remove spatial patterns. One alternative in this situation is to use two-dimensional smoothing [40]. However, in this case instead of (or as well as) smoothing over intensity levels, the smoothing is done with respect to the x and y coordinates of the array and it is unclear at this stage what the biological interpretability of this step is. It could well be that true biological variation is being removed, which is undesirable.

The majority of microarray experiments consist of more than one slide, and it is easily observed that there is more variation of measurements between slides than within slides. Most of this variation is due to technical aspects of both printing arrays and performing the actual experiments. In order to analyze a group of slides, most statistical methods assume that the slides have equal distributions of intensity levels; otherwise one slide might unfairly influence the results. The simplest way to deal with this issue is to either scale all the arrays so that they have equal variance or by adding a slide covariate to the model used to analyze the data (Fig. 4). Note that this type of normalization can also be conducted within slide, e.g. by print tip group, if necessary.

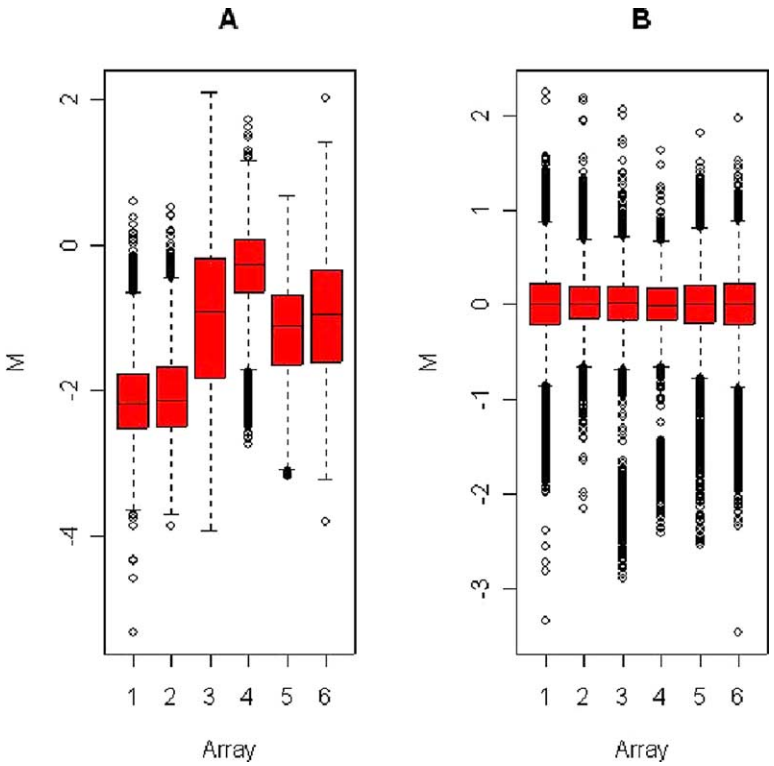

Fig. 4. Boxplots showing the distribution of M values in each of 6 arrays hybridized as part of the same experiment, before (left) and after (right) scale normalization. Data courtesy Prof. A.B. Smit, Department of Molecular and Cellular Neurobiology, Vrije Universiteit, Amsterdam, The Netherlands.

For all methods mentioned here, an important consideration is whether all or only some of the spots on the array(s) should be used for the normalization? Housekeeping genes have been shown to have nonconstant gene expression over all conditions, so it is most likely not a good idea to normalize using this set of genes. Use of control spots for normalization will depend on nature of these spots on the array, and their location. For example, if the control spots are only at the edges of the array, spatial differences cannot be adequately accounted for, meaning that normalization using these spots will most likely not remove all spatial irregularities. Specialist control spots, such as spiked controls, where expected expression intensities are known, are good but are not available on all arrays. The most important step, after carrying out normalization is to check the data visually, to make sure all artifacts have been removed from the data before more extensive analysis is conducted.

4. Inference, gene selection

One of the fundamental tasks of microarray data analysis is to identify genes that are regulated differently for *a priori* defined biologically relevant groups of samples. In this process, inference, two steps are crucial: definition of the quantity measuring differential expression, which enables us to rank the genes, and assessing statistical significance of the results. Currently, inference is performed by either permutation methods or model-based methods. These two approaches may be described as follows. Permutation methods rely on a test statistic which defines the quantity for differential expression. Its significance is assessed by comparing its observed value with the null-distribution. This null-distribution is usually obtained by permuting the sample labels simultaneously for all genes. Common statistics for differential expression are the t-statistic (two treatments case e.g. wild type vs. knockout) and F-statistic ($k > 2$ treatments case) and their nonparametric counterparts the Wilcoxon and Kruskal–Wallis statistics, which are more robust against outliers in the data (and hence against incorrect flagging). The model-based approach defines differential gene expression by a parameter in a statistical model. This model explains the observed data from several parameters and random noise or error. Methods to perform inference vary, but in any case they are critically dependent on distributional assumptions (e.g. normal) about the noise. Consider the case of two color arrays, where all samples are hybridized against a common reference and dye swaps are included. Concentrating on one gene, the log-ratio expression Y is modeled as:

$$Y = B + D + S + E,$$

where B is the basic expression, D reflects the dye-effect, S is the effect for the particular biological sample and E the normally distributed error. Then, the estimate of S is the model-based gene expression measure. In this particular case, inference may be performed by analysis of variance. More complex versions of this model are discussed in [18,42].

An interesting feature of some models is the fact that not only the mean expression is modeled, but also the standard error. There are two advantages in doing so: per gene, the estimate of a standard error may be more accurate, because it uses information from all the genes (rather than simply applying the basic formula to compute standard error from independent replicates) and errors may be propagated to estimate biological effects more accurately. The first can be best illustrated by the following: suppose that for gene A there are no technical replicates but 5 biological replicates, while for a large group of other genes technical replicates exist. Obviously, the biological replicates for gene A will include technical error as well, but when we compute the standard error (se) with the basic formula, the two errors are indistinguishable. Hence, the error is computed using 5 data points only. However, using an error model, for example with a multiplicative and additive error [30], allows one to obtain an estimate of the technical error in the gene A measurements as well, effectively using the data of the other genes with technical replicates.

To illustrate the latter advantage, i.e. propagation of the error, suppose there are $3 \times 2 = 6$ ratios for one particular gene: three biological replicates and two technical replicates per biological replicate. Now, suppose that the two technical replicates strongly disagree for the first biological replicate, but highly agree for the other two biological replicates. Propagation of the technical error implies that the first biological replicate receives less weight than the other two when the differential measure between the two conditions is computed. This is illustrated in Fig. 5.

An interesting development is to perform inference on groups of genes. Such a group would then have a common feature, e.g. all genes participate in the same pathway, and their definition would be based on other

data sources. One is then interested in a group-wise effect. Advantages over a gene-by-gene analysis are: increase of power, because one simply has more data for each group than for each gene and multiple testing corrections (to be discussed in the next section) are, if needed at all, less conservative, because the number of groups is usually much smaller than the number of genes. In [25] a permutation-based method called gene set enrichment analysis (GSEA) was proposed and it is shown that, in case of diabetes, one particular group of genes can be shown to have a differential effect, whereas none of the single genes are found to do so. It is not *a priori* clear how to measure group-wise effects. For example, assuming a simple common positive or negative effect may not be realistic in a pathway context, where usually negative feedbacks exist. If a set of genes is associated with different biological conditions, one does, however, expect more differential activity in both directions between the conditions [12].

Permutation methods may be too discrete when the number of biological replicates is small, especially when multiple comparisons are taken into account. For example, in a two treatment case with 4 biological samples per treatment, the smallest possible two-sided marginal p-value is $2/70 = 0.029$, which in most cases will be increased above the 0.05 level after applying a multiple testing correction. The situation improves with, say, 8 samples per treatment, when the smallest p-value equals 0.00031. When few biological replicates are available, assuming normal distributions and using a t-test may result in smaller p-values. Checking the validity of this assumption of normality (e.g. using a normal probability plot) is difficult for small studies, however if it holds for larger studies performed on the same platform, it may be reasonable to extend the conclusions to the smaller study.

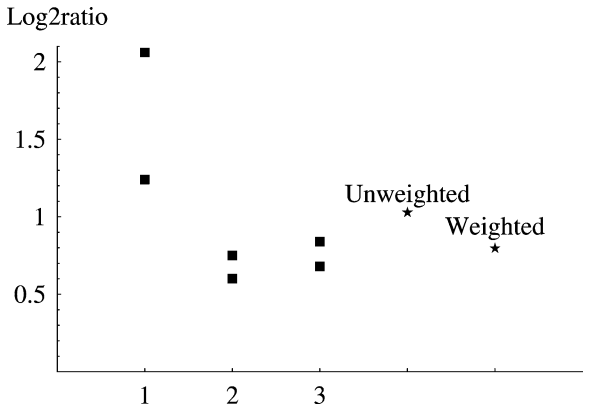

Fig. 5. Unweighted and weighted averages ($\star$) over three biological replicates (■). Weight per biological replicate is inversely related to the standard deviation between the two technical replicates. We observe that the unweighted average may be biased upwards due to the two inconsistent technical replicates of the first biological replicate.

5. Multiple testing

One of the strengths of microarray experiments, the ability to screen thousands of genes at the same time, has a downside as well. When performing statistical inference, severe corrections are needed with respect to common gene-by-gene analysis such as univariate t-tests. Consider a simulated experiment with 20,000 genes, two conditions (such as control vs. treatment) and 5 samples per condition. We assume that there is no differential expression at all for any gene. Naturally, biological and technical variation will occur. For simplicity, we assume this variation is the same for every measurement. We simulate this situation in the simplest way: each measurement is a random draw from a standard normal distribution (i.e. mean 0, variance 1). On the simulated data, we perform 20,000 two-sample t-tests. Figure 6 is a histogram of the p-values, around 1,000 of those being smaller than 0.05. Hence, using the threshold 0.05 we would mistakenly find 1,000 'significantly' expressed genes. This mistake is due to the multiplicity of the number of tests and hence so-called multiple testing corrections are necessary. The best-known, Bonferroni, means multiplication of each p-value by the number of tests. In microarray settings, with a large number of tests and small sample numbers this correction is often overly conservative and essentially useless. The Bonferroni correction controls the family-wise error (FWE): the probability that at least one gene is called significantly expressed while in reality it is not. A more powerful approach which controls

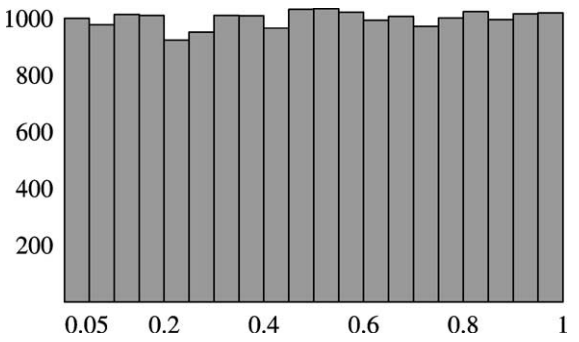

Fig. 6. Histogram of 20,000 p-values when all genes are not differentially expressed.

the FWE is the Westfall and Young [39] step-down method (implemented in Bioconductor). In microarray settings one might allow for more than one error (and hence hope to find more genes). Recently, extensions to the FWE control procedures to allow for k errors maximally have been developed [37].

An alternative to FWE control is control of the False Discovery Rate (FDR): the proportion falsely called genes of the total number of genes called. Benjamini and Hochberg [3] provide a simple control rule: multiply the univariate p-values by the number of genes and divide by the rank of the p-value. Another popular approach was introduced in [36], Significance Analysis of Microarrays (SAM), (available as an easy-to-use Excel add-in). In the context of a Bayesian hierarchical model, the FDR is typically controlled by imposing a mixture of distribution functions components, one of which represents the null-distribution (e.g. standard normal), and the others represent expression, on the statistic that measures differential gene expression. Such models were recently applied to breast cancer data [6].

When searching for (new) genes associated with a particular phenotype, multiple testing corrections commonly result in deceptively few 'significant' genes. A lot may be gained by restricting oneself *a priori* to a relatively small number of genes, usually those which are most promising based on biological knowledge. Then, the smaller number of tests leads to less severe corrections. As discussed in the previous section, use of sets of genes also reduces the multiplicity problem.

6. Clustering and classification

One of the major tasks commonly faced in microarray analysis is summarizing the large quantities of data into smaller, clearer components. Here we discuss two major techniques of multivariate analysis commonly applied to microarray data: unsupervised clustering and classification, the latter is sometimes referred to as supervised clustering or discriminant analysis. Unsupervised clustering groups the samples into unknown classes, whereas supervised clustering assigns new samples to a known class.

Clustering may be of interest for both genes or samples. Clustering of genes may be of use when trying to find genes in a common pathway, although the success could be limited. For example, use of correlation distance will cluster positively correlated genes, ignoring, for example, negative feedback loops. Clusters are often displayed by a tree, the branches of which could be arbitrarily swapped, so try not to be mislead by the graphical display of clusters. Before clustering one has to define a distance measure. Among others, the Euclidean distance (which in three dimensions or lower is simply the 'travelling distance' between two coordinates) and (Pearson) correlation are often used. The latter is especially useful for clustering of genes in time-course experiments. Software is abundant: besides specialized packages, all microarray data analysis packages, as well as most statistical packages contain several clustering procedures. The biological meaning of sample clusters is often shown by Kaplan–Meier survival plots for the two- (or three) main clusters. When survival differs significantly between the clusters, one may infer that the vector of the gene expression values has prognostic value. Sometimes, (part of) the sample clusters are shown to be biologically meaningful by considering common clinical features of the samples in one cluster.

When clustering samples, a difficult issue is: which genes to use? Ideally, one would like to use all available information and hence all genes. Clustering relies on genes that have discriminatory power: i.e. show very different expression levels over the samples. It is a fact that some genes may have many missing values, imputing of which may have an undesired 'anti-discriminatory' effect on clustering. Moreover, some genes correspond to many imprecise measurements. The latter may be coped with by introducing weighted clustering [44], which assigns relatively low weights to such genes. A useful and natural pre-processing step to clustering is principle components analysis (PCA), which is also available in most microarray analysis software packages. A principle component summarizes the entire vector of gene expression values into one number. The first principle component (PC) does this such that the variability between the samples according to the value of this component is maximized; the second maximizes the residual variability when accounted for the first PC and so on. As a set they maximize the explained variability between samples. Hence, these PCs may have a lot of discriminatory power for clustering analysis and it is in effect a weighted analysis, assigning more weight to genes showing large differences over the samples. We refer to [4] for an example. The PCs are sometimes called metagenes or supergenes, which might imply some biological meaning. However, inspection of the PC's will in most cases not support any biological interpretation.

Finally, we would like to note that clustering may critically depend on the quality of the samples (or arrays). It is not uncommon to find bad quality arrays ending up in one cluster, which is especially dangerous when one does not realize this and tries to assign biological meaning to the clusters. For a comparative review on various clustering methods, see [33].

Classification of different tumor types is very important in cancer diagnosis and drug discovery. Classification is a huge research area to which both the statistics and bioinformatics community have contributed. Rather than discussing all algorithms here in detail, we instead mention some of the errors and pitfalls commonly encountered. First of all, one might think that the group of most differential expressed genes is a good classifier. It will certainly have some discriminatory power, but in general many of those top genes will be highly correlated because of participation in the same pathway. That is, when making a classifier for tumors, one could include a lot of genes that act on cell proliferation, but the additional information decreases in the process of including those genes. Therefore, after including a few of those genes, one might obtain a better classifier by including less differentially expressed genes from other pathways.

An absolute crucial part of classification is cross-validation. In fact, it is easy to build a classifier which is absolutely perfect for the data set at hand, because one has so many 'predictors' (all the genes) and usually relatively few 'outcomes' (class label of the samples). Therefore, one has to guard oneself against overfitting. Leave-one-out cross-validation allows prediction of the probability of misclassification using the proposed classifier, which is essential to assess the classifier or simply as a risk calculation. If the number of samples is large enough, one may randomly split the samples into a learning set (used to build the classifier) and test set, which is used to assess the classifier. Repeating this procedure results in a Monte Carlo cross validation error estimate. Finally, especially if one is interested in producing, for example, diagnostic arrays with a limited number of genes, feature selection is an important issue. When performed, either externally (some genes might be *a priori* not interesting for this goal) or internally, usually by penalizing the number of genes in the classifier, it should be done on each of the test sets in the cross validation procedure separately to find the correct error rate of the entire procedure. Some of the classification methods, such as classification trees, automatically incorporate feature selection. We refer to [9] for an extensive overview, discussion and comparison of several classification algorithms as well as software options. Another useful overview with special emphasis on cancer classification is [24].

An interesting development is to merge gene expression data with Gene Ontology data. The Gene Ontology data, which describes known functional relationships between genes by a tree structure, are useful to reduce dimensionality of the data in a biologically very meaningful way. Classification method using GO-terms is discussed in [23]. Classification is often used for predicting categoric status (e.g. tumor type) of a sample. Ultimately, one might be interested in relating gene expression with a continuous measure like survival time or time to relapse. A combination of dimension reduction by PCA with a variation on a Cox regression model, which makes explicit use of survival time and censoring status is proposed in [21].

7. Pathways

Identifying which genes are differentially expressed in treated compared to normal samples is of course only the first step in trying to improve biological understanding. In what ways does the (non-)expression of those genes affect phenotype? Biologists are now also seeking the answers to these questions with the use of microarray data.

It could be assumed that genes which have similar expression patterns also have the same regulators. With this in mind, various groups have searched upstream regions of co-expressed genes in order to identify binding sites and gain more insight into genetic networks [10,14,15]. Another approach gaining in popularity is representational analysis. That is, of the genes which are differentially expressed, are more (or less) of them from one GO function class than would be expected by chance? If so, then this class of genes plays a significant biological role in the condition under investigation. Functional class scoring and GSEA are other examples of this type of approach.

Finally, and perhaps most ambitiously, there is growing interest in the use of expression data to construct biological networks. Using array data alone, Bayesian networks, Boolean networks and recently graphical Gaussian models have been proposed [11,22]. So far they have not proved very successful in reconstructing known networks from array data, even for simple eukaryotic organisms such as yeast. More recently, array data (such as time course, gene knock out series and RNAi) have been used in conjunction

with other databases and genetic information (known transcription factor binding sites, protein–protein interactions, DNA binding potentials etc) in the hope that this will improve the networks [13,26,28,32,38]. So far, with mixed success.

8. Literature overview

During the last few years a large number of books have appeared on microarray data analysis, both statistical books, which include details on models and algorithms, and descriptive books that aim to guide biologists in when to use what method. The latter usually to get across the main ideas behind certain methods and for solutions in 'standard' situations (e.g. control vs. treatment comparisons with many biological replicates), while the first may provide (less straightforward) solutions in other situations. We do not provide a complete list here, but just a number of books that we found useful: [2,27,34,35,41] for detailed statistical background and [8,19] for general background. Also, a variety of methods is reviewed in the supplement of *Nature Genetics* (2002), volume 32, pages 461–552.

The amount of software, both commercial and freeware, available for microarray analysis has exploded in recent years. When considering what software to use, it useful to consider the following issues:

- Data import: different image analysis packages give different file formats and, especially with large studies, it is most convenient when these files can be read in an automatic way.
- Specific packages versus comprehensive packages: does one want to perform one particular analysis in the best possible way, then specific packages are often most suitable. Comprehensive (and usually commercial) packages may not have all the options for particular modules, but allow easy transfer of results of one type of analysis to another (e.g. application of PCA to clustering).
- Most commercial packages are strong in visualization.
- Database programs (such as Rosetta Resolver) tend to be somewhat 'over standardized' for analysis means and do not always allow arbitrary experimental designs.
- Freeware is wonderful, but often not debugged.
- Standard statistical software (such as S-Plus, SAS or matlab) is usually extensively debugged.

Packages based on the language 'R', such as those in the Bioconductor project (see www.bioconductor.org), seem to have become the standard within the statistical community. User-friendliness varies among the packages available which are written by different authors. We had positive experiences with limma and the R-package maanova, which do normalization and inference (plus multiple testing corrections). Some commercial packages, like Spotfire, provide tools to run R-scripts within the package. We cannot list all available software here, but refer to the following microarray software sites: http://genome-www5.stanford.edu/ and, for a extensive list and short descriptions of several packages: http://www.cs.tcd.ie/Nadia.Bolshakova/softwaretotal.html.

9. Conclusions

Microarray data analysis is far from easy and the amount of effort a proper analysis requires is often underestimated. It is difficult to standardize all the analysis steps described in this paper as different data sets may need different approaches. Much can be gained by thinking before carrying out the experiment. Limit the number of hypotheses: do not try to solve 5 questions with a budget for only 6 arrays. Prioritize the hypotheses and design the experiment that suits the most important question best. It is wise to approach the analysis with the same philosophy as for the experiment itself: check after every step. Do the results confirm your knowledge or intuition, for example, do you get a nice straight line after normalization? Finally, validating the microarray results using other techniques (e.g. qPCR or Northern Blot) and database information is essential.

Acknowledgements

We thank A.B. Smit and F.J. Stam for providing the microarray data. This work was supported in part by a CLS grant from the Netherlands Organisation for Scientific Research (NWO) (to N.J.A.) and by the Dutch BRICKS consortium (to M.A.v.d.W.).

References

[1] A.A. Alizadeh, M.B. Eisen, R.E. Davis, C. Ma, I.S. Lossos, A. Rosenwald, J.C. Boldrick, H. Sabet, T. Tran, X. Yu, J.I. Powell, L. Yang, G.E. Marti, T. Moore, J. Hudson, Jr, L. Lu, D.B. Lewis, R. Tibshirani, G. Sherlock, W.C. Chan, T.C. Greiner, D.D. Weisenburger, J.O. Armitage, R. Warnke,

R. Levy, W. Wilson, M.R. Grever, J.C. Byrd, D. Botstein, P.O. Brown and L.M. Staudt, Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling, *Nature* **403** (2000), 503–511.

[2] P. Baldi and G.W. Hatfield, *DNA Microarrays and Gene Expression, from Experiments to Data Analysis and Modeling*, Cambridge University Press, 2002.

[3] Y. Benjamini and Y. Hochberg, Controlling the false discovery rate: A practical and powerful approach to multiple testing, *J. Roy. Statist. Soc. B* **57** (1995), 289–300.

[4] J.R. Bleharski, H. Li, C. Meinken et al., Use of genetic profiling in leprosy to discriminate clinical forms of the disease, *Science* **301** (2003), 1527–1530.

[5] B.M. Bolstad, R.A. Irizarry, M. Astrand and T.P. Speed, A comparison of normalization methods for high density oligonucleotide array data based on variance and bias, *Bioinformatics* **19** (2003), 185–193.

[6] P. Broët, A. Lewin, S. Richardson, C. Dalmasso and H. Magdelenat, A mixture model-based strategy for selecting sets of genes in multiclass response microarray experiments, *Bioinformatics*, in press.

[7] G.A. Churchill, Fundamentals of experimental design for cDNA microarrays, *Nat. Genet.* **32** (2002), 490–495.

[8] S. Draghici, *Data Analysis Tools for DNA Microarrays*, Chapmann-Hall, 2003.

[9] S. Dudoit and J. Fridlyand, *Classification in Microarray Experiments*, Chapman and Hall, 2003, pp. 93–158.

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

[11] N. Friedman, M. Linial, I. Nachman and D. Pe'er, Using Bayesian networks to analyze expression data, *J. Comput. Biol.* **7** (2000), 601–620.

[12] J.J. Goeman, S.A. van de Geer, F. de Kort and H.C. van Houwelingen, A global test for groups of genes: testing association with clinical outcome, *Bioinformatics* **20** (2004), 93–99.

[13] A.J. Hartemink, D.K. Gifford, T.S. Jaakkola and R.A Young, Combining location and expression data for principled discovery of genetic regulatory network models, in: *Pacific Symposium on Biocomputing*, 2002, pp. 437–449.

[14] J.D. Hughes, P.W. Estep, S. Tavazoie and G.M. Church, Computational identification of cis-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*, *J. Mol. Biol.* **296** (2000), 1205–1214.

[15] L.J. Jensen and S. Knudsen, Automatics discovery of regulatory patterns in promotor regions based on whole cell expression data and functional annotation, *Bioinformatics* **16** (1999), 326–333.

[16] T.K. Jenssen, W.P. Langaas, M. Kuo, B. Smith-Sørensen, O. Myklebost and E. Hovig, Analysis of repeatability in spotted cDNA microarrays, *Nucleic Acids Res.* **30** (2002), 3235–3244.

[17] M.K. Kerr and G.A. Churchill, Experimental design for gene expression microarrays, *Biostatistics* **2** (2001), 183–201.

[18] M.K. Kerr, M. Martin and G.A. Churchill, Analysis of variance for gene expression microarray data, *J. Comput. Biol.* **7** (2000), 819–837.

[19] S. Knudsen, *A Biologist's Guide to Analysis of DNA Microarray Data*, Wiley, 2002.

[20] C. Li, G.C. Tseng and W.H. Wong, *Model-Based Analysis of Oligonucleotide Arrays and Issues in cDNA Microarray Analysis*, Chapman and Hall, 2003, pp. 1–34.

[21] H. Li and J. Gui, Partial Cox regression analysis for high-dimensional microarray gene expression data, *Bioinformatics* **20**(Suppl. 1) (2004), i208–i215.

[22] S. Liang, R. Fuhrman and R. Somogyi, Reveal, a general reverse engineering algorithm for inference of genetic network architectures, in: *Pacific Symposium on Biocomputing*, 1998, pp. 18–29.

[23] C. Lottaz, StAM: Structured analysis of microarray data. Max Planck Institute for molecular genetics, http://compdiag.molgen.mpg.de/research/project_stam.shtml, 2004.

[24] Y. Lu and J. Han, Cancer classification using gene expression data, *Information Systems* **28** (2003), 243–268.

[25] V.M. Mootha, C.M. Lindgren, K. Eriksson et al., PGC-1α-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes, *Nat. Genet.* **34** (2003), 267–273.

[26] I. Nachman, A. Regev and N. Friedman, Inferring quantitative models of regulatory networks from expression data, *Bioinformations* **20** (2004), I248–I256.

[27] G. Parmigiani, E.S. Garett, R.A. Irizarry and S.L. Zeger, *The Analysis of Gene Expression Data*, Springer, 2003.

[28] D. Pe'er, A. Regev, G. Elidan and N. Friedman, Inferring subnetworks from perturbed expression profiles, *Bioinformatics* **1** (2001), 1–9.

[29] J. Quackenbush, Microarray data normalization and transformation, *Nat. Genet.* **32** (2002), 496–501.

[30] D.M. Rocke and B. Durbin, A model for measurement error for gene expression arrays, *J. Comput. Biol.* **8** (2001), 557–569.

[31] M. Schena, D. Shalon, R. Heller, A. Chai, P.O. Brown and R.W. Davis, Parallel human genome analysis: microarray-based expression monitoring of 1000 genes, *Proc. Nat. Acad. Sci. USA* **93** (1996), 10614–10619.

[32] E. Segal, H. Wang and D. Koller, Discovering molecular pathways from protein interaction and gene expression data, *Bioinformatics* **19** (2003), I264–I272.

[33] R. Shamir and R. Sharan, Algorithmic approaches to clustering gene expression data, in: *Current Topics in Computational Biology*, Y. Xu T. Jiang, T. Smith and M.Q. Zhang, eds, MIT Press, 2001.

[34] T. Speed et al., *Statistical Analysis of Gene Expression Microarray Data*, Chapman and Hall, 2003.

[35] M.L. Ting Lee, *Analysis of Microarray Gene Expression Data*, Springer, 2004.

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

[37] M.J. Van der Laan, S. Dudoit and K.S. Pollard, Augmentation procedures for control of the generalized family-wise error rate and tail probabilities for the proportion of false positives, *Statistical Applications in Genetics and Molecular Biology* **3**(1) (2004), Article 15.

[38] W. Wang, J.M. Cherry, D. Botstein and H. Li, A systematic approach to reconstructing transcription networks in *Saccharomyces cerevisiae*, *Proc. Natl. Acad. Sci. USA* **99** (2002), 16893–16898.

[39] P.H. Westfall and S.S. Young, *Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment*, Wiley, New York, 1993.

[40] D.L. Wilson, M.J. Buckley, C.A. Helliwell and I.W. Wilson, New normalization methods for cDNA microarray data, *Bioinformatics* **19** (2003), 1325–1332.

[41] E. Wit and J. McClure, *Statistics for Microarrays: Design, Analysis and Inference*, Wiley, 2004.

[42] R.D. Wolfinger, G. Gibson, E.D. Wolfinger, H. Bennett, P. Bushel, C. Afshari and R.S. Paules, Assessing gene significance from cDNA microarray expression data via mixed models, *J. Comput. Biol.* **8** (2001), 625–637.

[43] Y.H. Yang and T. Speed, Design issues for cDNA microarray experiments, *Nat. Rev. Genet.* **3** (2002), 579–588.

[44] K.Y. Yeung, M. Medvedovic and R.E. Bumgarner, Clustering gene-expression data with repeated measurements, *Genome Biology* **4** (2003), R4.