14

Analysis of Differential Gene Expression Studies

D. Scholtens and A. von Heydebreck

Abstract

In this chapter, we focus on the analysis of differential gene expression studies. Many microarray studies are designed to detect genes associated with different phenotypes, for example, the comparison of cancer tumors and normal cells. In some multifactor experiments, genetic networks are perturbed with various treatments to understand the effects of those treatments and their interactions with each other in the dynamic cellular network. For even the simplest experiments, investigators must consider several issues for appropriate gene selection. We discuss strategies for gene-at-a-time analyses, nonspecific and meta-data driven prefiltering techniques, and commonly used test statistics for detecting differential expression. We show how these strategies and statistical tools are implemented and used in Bioconductor. We also demonstrate the use of factorial models for probing complex biological systems and highlight the importance of carefully coordinating known cellular behavior with statistical modeling to make biologically relevant inference from microarray studies.

14.1 Introduction

Microarray technology is used in a wide variety of settings for detecting differential gene expression. Classic statistical issues such as appropriate test statistics, sample size, replicate structure, statistical significance, and outlier detection enter into the design and analysis of gene expression studies. Adding to the complexity is the fact that the number of samples I in a microarray experiment is inevitably much less than the number of genes J under investigation and that J is often on the scale of tens of thousands,

thus creating a tremendous multiple testing burden (see Chapter 15 for further discussion). Investigators must ensure that the experimental design gives access to unambiguous tests of the key substantive hypotheses. This is a challenging task in the complex, dynamic cellular network. We begin our discussion in Section 14.2 by examining general issues in differential expression analysis relevant to most microarray experiments, illustrating these principles with case studies of the ALL and kidpack data in Sections 14.2.1 and 14.2.2. We then examine multifactor models in Section 14.3 with a case study of the estrogen data in Section 14.3.1.

14.2 Differential expression analysis

Fundamental to the task of analyzing gene expression data is the need to identify genes whose patterns of expression differ according to phenotype or experimental condition. Gene expression is a well coordinated system, and hence measurements on different genes are in general not independent. Given more complete knowledge of the specific interactions and transcriptional controls, it is conceivable that meaningful comparisons between samples can be made by considering the joint distribution of specific sets of genes. However, the high dimension of gene expression space prohibits a comprehensive exploration, while the fact that our understanding of biological systems is only in its infancy means that in many cases we do not know which relationships are important and should be studied. In current practice, differential expression analysis will therefore at least start with a gene-by-gene approach, ignoring the dependencies between genes.

A simple approach is to select genes using a fold-change criterion. This may be the only possibility in cases where no, or very few replicates, are available. An analysis solely based on fold change however does not allow the assessment of significance of expression differences in the presence of biological and experimental variation, which may differ from gene to gene. This is the main reason for using statistical tests to assess differential expression. Generally, one might look at various properties of the distributions of a gene's expression levels under different conditions, though most often location parameters of these distributions, such as the mean or the median, are considered. One may distinguish between parametric tests, such as the t-test, and non-parametric tests, such as the Mann-Whitney test or permutation tests. Parametric tests usually have a higher power if the underlying model assumptions, such as normality in the case of the t-test, are at least approximately fulfilled. Non-parametric tests do have the advantage of making less stringent assumptions on the data-generating distribution. In many microarray studies however, a small sample size leads to insufficient power for non-parametric tests. A pragmatic approach in these

situations is to employ parametric tests, but to use the resulting p-values cautiously to rank genes by their evidence for differential expression.

When performing statistical analysis of microarray data, an important question is determining on which scale to analyze the data. Often the logarithmic scale is used in order to make the distribution of replicated measurements per gene roughly symmetric and close to normal. A variance-stabilizing transformation derived from an *error model* for microarray measurements (see Chapter 1) may be employed to make the variance of the measured intensities independent of their expected value (Huber et al., 2002). This can be advantageous for gene-wise statistical tests that rely on variance homogeneity, because it will diminish differences in variance between experimental conditions that are due to differences in the intensity level – however of course differences in variance between conditions may also have gene-specific biological reasons, and these will remain untouched.

One or two group t-test comparisons, multiple group ANOVA, and more general trend tests are all instances of linear models that are frequently used for assessing differential gene expression. As a parametric method, linear modeling is subject to the caveats discussed above, but the convenient interpretability of the model parameters often makes it the method of choice for microarray analysis. Due to the aforementioned lack of information regarding coregulation of genes, linear models are generally computed for each gene separately. When the lists of genes of interest are identified, investigators can hopefully begin to study their coordinated regulation for more sophisticated modeling of their joint behavior.

The approach of conducting a statistical test for each gene is popular, largely because it is relatively straightforward and a standard repertoire of methods can be applied. However, the approach has a number of drawbacks: most important is the fact that a large number of hypothesis tests is carried out, potentially leading to a large number of falsely significant results. *Multiple testing* procedures allow one to assess the overall significance of the results of a family of hypothesis tests. They focus on specificity by controlling type I (false positive) error rates such as the *family-wise error rate* or the *false discovery rate* (Dudoit et al., 2003). This topic is covered in detail in Chapter 15. Still, multiple hypothesis testing remains a problem, because an increase in specificity, as provided by p-value adjustment methods, is coupled with a loss of sensitivity, that is, a reduced chance of detecting true positives. Furthermore, the genes with the most drastic changes in expression are not necessarily the "key players" in the relevant biological processes. This problem can only be addressed by incorporating prior biological knowledge into the analysis of microarray data, which may lead to focusing the analysis on a specific set of genes. Also if such a biologically motivated preselection is not feasible, the number of hypotheses to be tested can often be reasonably reduced by non-specific filtering procedures, discarding, e.g., genes with consistently low intensity values or low variance across the samples. This is especially relevant in the case of genome-wide

arrays, as often only a minority of all genes will be expressed at all in the cell type under consideration.

Many microarray experiments involve only few replicates per condition, which makes it difficult to estimate the gene-specific variances that are used, e.g., in the t-test. Different methods have been developed to exploit the variance information provided by the data of all genes (Baldi and Long, 2001; Tusher et al., 2001; Lönnstedt and Speed, 2002; Kendziorski et al., 2003). In the limma package, an Empirical Bayes approach is implemented that employs a global variance estimator s_0^2 computed on the basis of all genes' variances. The resulting test statistic is a moderated t-statistic, where instead of the single-gene estimated variances s_g^2, a weighted average of s_g^2 and s_0^2 is used. Under certain distributional assumptions, this test statistic can be shown to follow a t-distribution under the null hypothesis with the degrees of freedom depending on the data (Smyth, 2004).

In the following examples, we demonstrate the use of Bioconductor packages, especially multtest and limma, to identify differentially expressed genes.

14.2.1 Example: ALL data

In this example, we consider a subset of the ALL data representing 79 samples from patients with B-cell acute lymphoblastic leukemia that were investigated using HG-U95Av2 Affymetrix GeneChip arrays (Chiaretti et al., 2004). The probe-level data were preprocessed using RMA (Irizarry et al., 2003b), described in Chapter 2, to produce log (base 2) expression measurements. Of particular interest is the comparison of samples with the BCR/ABL fusion gene resulting from a translocation of the chromosomes 9 and 22 with samples that are cytogenetically normal. In the following code chunk, we load the data and define the subset of samples we are interested in – 37 BCR/ABL samples and 42 normal samples (labeled NEG). The *exprSet* object `eset` contains the relevant data.

```
> library("ALL")
> data(ALL)
> pdat <- pData(ALL)
> subset <- intersect(grep("^B", as.character(pdat$BT)),
+     which(pdat$mol %in% c("BCR/ABL", "NEG")))
> eset <- ALL[, subset]
```

Many of the genes represented by the 12625 probesets on the array are not expressed in B-cell lymphocytes (either in their normal condition or in any of the disease states being considered), which are the cells that were measured in this experiment. Hence the probesets for these genes can, and should, be removed from the analysis. Furthermore, we want to discard probesets with a low variability across all samples. In the next code chunk, we require expression measurements to be above 100 fluorescence units in

at least 25% of the samples, and the interquartile range (IQR) across the samples on the log base 2 scale to be at least 0.5. This non-specific filtering is accomplished with functions from the package genefilter.

```
> library("genefilter")
> f1 <- pOverA(0.25, log2(100))
> f2 <- function(x) (IQR(x) > 0.5)
> ff <- filterfun(f1, f2)
> selected <- genefilter(eset, ff)
> sum(selected)
```

```
[1] 2391
```

```
> esetSub <- eset[selected, ]
```

We are left with 2391 probesets for further analysis. Using the multtest package, we perform a permutation test for equality of the mean expression levels in the two groups for each of these probesets. By default, the function `mt.maxT` computes Welch t-statistics, which allow for unequal variances in the two groups. The number of permutations `B` determines the granularity of the permutation p-values. Depending on the multiple testing procedure to be applied, the user may have to choose a value of `B` that is considerably larger than the number of tests being performed.

```
> cl <- as.numeric(esetSub$mol == "BCR/ABL")
> resT <- mt.maxT(exprs(esetSub), classlabel = cl,
+     B = 10000)
> ord <- order(resT$index)
> rawp <- resT$rawp[ord]
> names(rawp) <- geneNames(esetSub)
```

Figure 14.1 shows the histogram of unadjusted permutation p-values, as given by the vector `rawp`. The high proportion of small p-values suggests that a substantial fraction of the genes are differentially expressed between the two groups. In order to control the family-wise error rate (FWER), that is, the probability of at least one false positive in the set of significant genes, we have used the permutation-based maxT-procedure of Westfall and Young (Westfall and Young, 1993), as implemented in the function `mt.maxT`. We obtain 18 genes with an adjusted p-value below 0.05:

```
> sum(resT$adjp < 0.05)
```

```
[1] 18
```

A comparison of this number to the height of the leftmost bar in the histogram suggests that we may be missing a large number of differentially expressed genes. The FWER is a very stringent criterion, and in some microarray studies, only few genes may be significant in this sense, even if many more are truly differentially expressed. A more liberal criterion is provided by the false discovery rate (FDR), that is, the expected proportion of false positives among the genes that are called significant. We use the

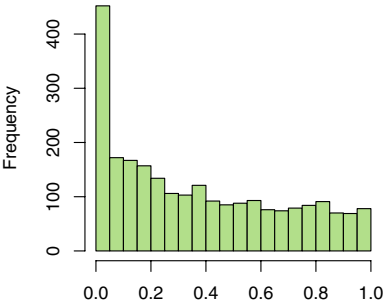

Figure 14.1. Histogram of p-values for the gene-by-gene comparison between BCR/ABL positive and cytogenetically normal leukemias.

procedure of Benjamini and Hochberg (1995) as implemented in multtest to control the FDR at a level of 0.05, which leaves us with 102 significant genes (note however that this procedure makes certain assumptions on the dependence structure between genes):

```
> res <- mt.rawp2adjp(rawp, proc = "BH")
> sum(res$adjp[, "BH"] < 0.05)

[1] 102
```

Effects of non-specific filtering

As indicated above, the aim of non-specific filtering is to remove genes that, e.g., due to their low overall intensity or variability, are unlikely to carry information about the phenotypes under investigation. The researcher will be interested in keeping the number of tests as low as possible while keeping the interesting genes in the selected subset.

If the truly differentially expressed genes are overrepresented among those selected in the filtering step, the FDR associated with a certain threshold of the test statistic will be lowered due to the filtering. This appears plausible for two commonly used global filtering criteria: *Intensity-based filtering* aims to remove genes that are not expressed at all in the samples studied, and therefore cannot be differentially expressed. Also concerning the *variability across samples*, a higher overall variance of the differentially expressed genes may be expected, because their between-class variance adds to their within-class variance.

To investigate these presumed effects, we compare the scores for intensity and variability that we used in the beginning for gene selection with

the absolute values of the t-statistic, which we now compute for all 12625 probesets.

```
> IQRs <- esApply(eset, 1, IQR)
> intensityscore <- esApply(eset, 1, function(x) quantile(x,
+     0.75))
> abs.t <- abs(mt.teststat(exprs(eset), classlabel = cl))
```

The result is shown in Figure 14.2. Gene selection by the interquartile range (IQR) indeed seems to lead to a higher concentration of differentially expressed genes, whereas for the intensity-based criterion, the effect is less pronounced.

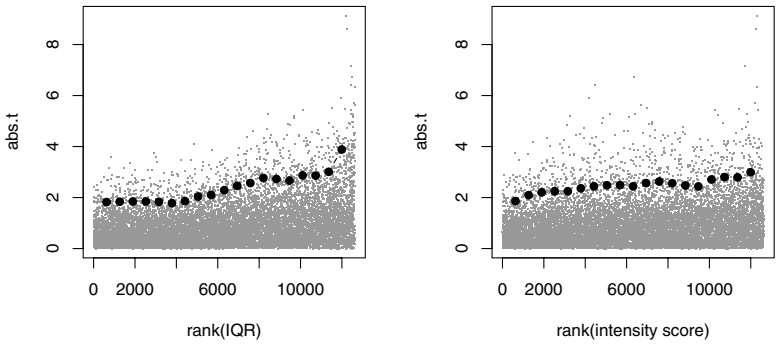

Figure 14.2. Plots of the absolute values of the t-statistic (y-axis) against the ranks of the values of the two filtering criteria: left, interquartile range (IQR), right, overall intensity score. The larger dark dots indicate the 95%-quantiles of the absolute value of the t-statistic computed for moving windows along the x-axis.

Using Gene Ontology data

A source of valuable biological data that is easily accessible through Bioconductor software is the Gene Ontology (GO). It is known that many of the effects due to the BCR/ABL translocation are mediated by tyrosine kinase activity. It will therefore be of interest to examine genes that are known to have tyrosine kinase activity. The term `GO:0004713` from the *molecular function* portion of the GO hierarchy refers to `protein-tyrosine kinase activity`. We can obtain all Affymetrix probesets that are annotated at that node, either directly or by inheritance, using the following command.

```
> tykin <- unique(lookUp("GO:0004713", "hgu95av2",
+     "GO2ALLPROBES"))
> length(tykin)
```

```
[1] 352
```

We see that 352 probesets are annotated at this particular term, 48 of which were selected by our non-specific filtering step. We focus our attention on these 48 probesets and repeat the permutation t-test analysis. In the analysis of the GO-filtered data, 6 probesets have FWER-adjusted p-values less than 0.05. They are printed below, together with the adjusted p-values from the first analysis that involved 2391 genes.

```
[1] "GO analysis"

40480_s_at    1635_at  1636_g_at   39730_at  2039_s_at
    0.0001     0.0001     0.0001     0.0001     0.0005
  36643_at
    0.0286

[1] "All Genes"

   1635_at  1636_g_at   39730_at 40480_s_at  2039_s_at
    0.0001     0.0001     0.0001     0.0015     0.0149
  36643_at
    0.4691
```

Due to the reduced number of tests in the analysis focused on tyrosine kinases, we are left with more significant genes after correcting for multiple testing. For instance, the probeset `36643_at`, which corresponds to the gene DDR1, was not significant in the unfocused analysis, but would be if instead the investigation was oriented toward studying tyrosine kinases.

14.2.2 Example: Kidney cancer data

The kidpack package contains gene expression data from 74 renal cell carcinoma (RCC) patient biopsy samples, which were measured on two-color cDNA arrays together with a common reference sample. The data set is described in detail in the Appendix A.1.2 and in Sültmann et al. (2005). The RCC samples belong to three different histological types, clear cell (ccRCC), papillary (pRCC) and chromophobe (chRCC):

```
> pdat <- pData(esetSpot)
> table(pdat$type)

ccRCC chRCC  pRCC
   52     9    13
```

In the following, we illustrate how the differences in gene expression between these types can be investigated using the limma package (see also Chapter 23 for a more detailed description of limma). We are going to fit a linear model to the expression levels of each gene. limma expects the model to be specified by the *design matrix*, which can either be defined directly or be constructed from a formula via the function `model.matrix`, which is what we do here:

```
> design <- model.matrix(~-1 + factor(pdat$type))
> colnames(design) <- c("ccRCC", "chRCC", "pRCC")
```

This simple design matrix corresponds to the following parametrization:

$$y_{ik} = \alpha_k + \epsilon_{ik} \quad i = 1, 2, \ldots, n_k; \quad k = 1, 2, 3,$$

where k indicates the tumor type and i the individual samples. Note that the model is parameterized without an intercept term, and the estimated coefficients $\hat{\alpha}_k$ from a least squares fit are the mean expression values for the three cancer types.

To exploit the information of replicate measurements of each cDNA clone, limma allows fitting linear models to the spot intensities taking the correlation between replicate spots into account (Smyth et al., 2005). First, the correlation between replicate spots is estimated for each gene separately with restricted maximum likelihood (REML) based on a mixed effects linear model. An overall estimate of the correlation between replicates is computed as a robust average of the individual correlations on the hyperbolic arc tangent scale (atanh), and this overall estimate is then used when fitting a linear model for each gene. The same procedure can be applied in the case of several hybridizations (technical replicates) per cell or tissue sample (biological replicate). In our case, we estimate the correlation between the two replicate spots per clone (argument `ndups`). The 4224 different clones are listed in separate row blocks in the expression data matrix, hence their `spacing` is 4224:

```
> dupcor <- duplicateCorrelation(exprs(esetSpot),
+     design = design, ndups = 2, spacing = 4224)
> fit <- lmFit(esetSpot, design = design, ndups = 2,
+     spacing = 4224, correlation = dupcor$cor)
```

```
> dupcor$cor
```

```
[1] 0.407
```

By default, `lmFit` fits a linear model by the least squares method, but it also allows robust regression. We are now interested in the expression differences between any two of the cancer types. For this purpose, we set up a *contrast matrix* whose columns represent the pairwise differences between the model coefficients. With the function `contrast.fit`, we can compute estimated coefficients and standard errors for these contrasts from our original model fit:

```
> contrast.matrix <- makeContrasts(ccRCC - chRCC,
+     ccRCC - pRCC, chRCC - pRCC, levels = design)
> contrast.matrix
```

```
      ccRCC - chRCC ccRCC - pRCC chRCC - pRCC
ccRCC             1            1            0
chRCC            -1            0            1
pRCC              0           -1           -1
```

```
> fit2 <- contrasts.fit(fit, contrast.matrix)
```

Moderated t-statistics for these contrasts, where the gene-specific variances are augmented with a global variance estimator computed from the data of all genes, are obtained with the function `eBayes`:

```
> fit3 <- eBayes(fit2)
```

The `topTable` function produces a table of the top ranking genes, sorted by default by their log-odds for differential expression (see below). Here we show the output of `topTable` for the third contrast, referring to the comparison of chRCC and pRCC.

```
> topTable(fit3, coef = 3, n = 8, adjust.method = "fdr")
```

```
         ID     M       A     t  P.Value    B
2600 321496  2.68 -0.1154  18.5 1.12e-37 82.7
2729 502969  1.88 -0.1703  13.6 6.45e-25 53.4
1804 133812  1.81 -0.5036  13.3 3.00e-24 51.5
2859 725766  1.92 -0.1276  12.9 1.69e-23 49.5
3734 306257 -1.53  0.1353 -12.4 3.84e-22 46.3
1879 357297  1.36 -0.3215  11.7 3.38e-20 41.7
1905 774064  1.74 -0.4917  11.4 1.15e-19 40.4
2750 738532  1.37  0.0461  11.3 3.47e-19 39.2
```

For the column `P.value`, different methods to adjust the p-values for multiple testing can be chosen, which allow to control the family-wise error rate or the false discovery rate. Here we have chosen the FDR-based p-value adjustment according to Benjamini and Hochberg (1995). Further columns produced by `topTable` contain for each gene an identifier `Name` (in our case the Image ID of the respective cDNA clone), the estimated contrast coefficient `M`, the average expression value across all samples `A`, the moderated t-statistic `t`, and the log-odds for differential expression `B`, corresponding to a Bayesian interpretation of the moderated t-statistic. The interpretation of the values of `M` and `A` depends on the nature of the data used as input for `lmFit`. In our case, the column `M` contains expression differences on a generalized natural log scale relative to a common reference sample, and the values of `A` do not refer to absolute intensities but are given by the average of a gene's generalized log-ratio values with respect to the reference sample across all chips.

When testing different contrasts per gene simultaneously, the issue of *multiple comparisons* arises, that is, it is of interest to evaluate the significance of each single contrast in the light of the whole set of contrasts. The limma function `decideTests` allows the identification of significant test results in the sense of *multiple testing* across genes, as well as in the sense of *multiple comparisons* across contrasts. For the latter, the following approach is pursued with the argument `method="nestedF"`: The moderated t-statistic for a particular contrast is called significant at a certain level α (resulting from multiple testing adjustment across genes) if the moderated

F-test for that gene is still significant at level α when setting all the larger t-statistics for that gene to the same absolute value as the t-statistic in question. The function `decideTests` yields a matrix, where for each gene each contrast is marked as non-significant (zero), significantly positive (one), or significantly negative (minus one). In our example, we want to know how many genes are differentially expressed when fixing the significance level α of the moderated F-test so that it corresponds to a FDR of 0.05:

```
> clas <- decideTests(fit3, method = "nestedF",
+     adjust.method = "fdr", p = 0.05)
> colSums(abs(clas))

ccRCC - chRCC  ccRCC - pRCC  chRCC - pRCC
         1243           981           931
```

To assess the effect of using the single spot measurements opposed to the commonly used averaging across duplicate spots, we compare the results to those of an analogous analysis based on a data matrix `datAvDup` where the expression values of duplicate spots have been averaged.

```
> nclones <- 4224
> datAvDup <- (exprs(esetSpot)[1:nclones, ] +
+     exprs(esetSpot)[nclones + 1:nclones, ])/2
> fitAvDup <- lmFit(datAvDup, design = design)
> fit2AvDup <- contrasts.fit(fitAvDup, contrast.matrix)
> fit3AvDup <- eBayes(fit2AvDup)
```

The comparison of the resulting p-values (again for the comparison of chRCC and pRCC) suggests that the spot-wise analysis yields higher power (Figure 14.3).

14.3 Multifactor experiments

Multifactor microarray experiments often involve the application of treatments in combination to model organisms such as genetically identical cell lines or mice. The equal reference point from which these experiments start theoretically limits naturally occurring interindividual variability, thus allowing differential gene expression to be attributed to the treatments or experimental conditions under investigation. Frequently, these experiments are designed to investigate the perturbation of genetic networks by various combinations of treatments, thus allowing the initial steps of genetic network reconstruction. In factorial designs, effects of the treatments and their interactions can be conveniently quantified in a linear model. As long as the contrasts of interest are specified with careful accounting for the transcription and translation mechanisms affected by the treatments, investigators can often assign very meaningful biological interpretations to their results.

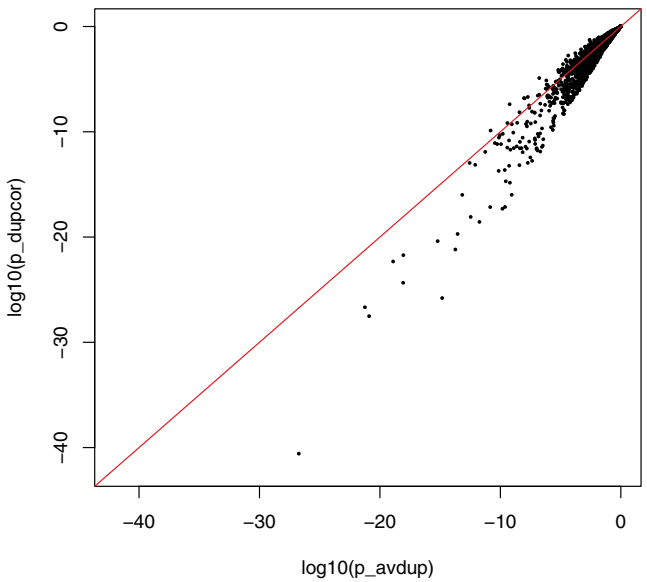

Figure 14.3. Comparison of base 10 logarithms of p-values for the comparison between chRCC and pRCC. x-axis: analysis based on average expression values across duplicate spots, y-axis: spot-wise analysis incorporating correlation between duplicate spots.

One significant difficulty with linear modeling in the microarray setting is model checking. Studentized residuals from the classic linear modeling paradigm are often inappropriate in designs with only a few replicates due to the large number of linear dependencies relative to the number of residuals. Specialized algorithms are often useful for very small designs; in Section 14.3.1, we discuss a technique for outlier detection in a factorial experiment with just two replicates. The development of multivariate permutation tests for the high-throughput setting would help alleviate this problem (Pesarin, 2001).

Multifactor linear models have been used for a variety of purposes in microarray studies. In addition to identifying differentially expressed genes due to treatments applied in combination, linear models have been very useful for data preprocessing of cDNA microarrays (see Chapter 4). In the estrogen example, we illustrate the interpretability of multifactor linear models for single channel arrays; the results extend naturally to two-color competitive hybridization platforms. We use the limma package for our

analysis, but factDesign and daMA are also available for the analysis of factorial designed microarray experiments.

14.3.1 Example: Estrogen data

The package estrogen contains 8 Affymetrix HG-U95Av2 CEL files from an experiment involving breast cancer cells. We first perform quantile normalization and calculate expression estimates using RMA (Irizarry et al., 2003b).

```
> library("estrogen")
> library("limma")
> library("hgu95av2cdf")
> datadir <- system.file("extdata", package = "estrogen")
> targets <- readTargets("phenoData.txt", path = datadir,
+     sep = "")
> covdesc <- list("present or absent", "10 or 48 hours")
> names(covdesc) <- names(targets)[-1]
> pdata <- new("phenoData", pData = targets[, -1],
+     varLabels = covdesc)
> rownames(pData(pdata)) <- targets[, 1]
> gc()
> esAB <- ReadAffy(filenames = file.path(datadir,
+     targets$filename), phenoData = pdata)
> esEset <- rma(esAB)
```

This collection of eight arrays is a subset of 32 arrays from a 2^4 factorial experiment with two replicates for each treatment condition on an estrogen receptor positive (ER+) breast cancer cell line, the complete analysis of which is discussed in Scholtens et al. (2004). Upon binding to estrogen, the estrogen receptor (ER) acts as a transcription factor for specific genes, either stimulating or repressing their expression and causing a host of downstream effects. The investigators were interested in identifying primary and secondary targets of estrogen in these cells, and noting any changes in mRNA transcript behavior for the targets over time. After serum starvation of all eight samples, four samples were exposed to estrogen and then harvested for microarray analysis after 10 hours for two samples and 48 hours for the other two. The remaining four samples were left untreated and harvested after 10 hours for two samples, and 48 hours for the other two. An *exprSet* named `esEset` contains expression levels for 12,625 probesets for the 8 samples described above, as well as the corresponding *phenoData* that specify the 2^2 factorial design.

```
> esEset

Expression Set (exprSet) with
        12625 genes
        8 samples
```

```
                phenoData object with 2 variables and 8 cases
        varLabels
               estrogen: present or absent
               time.h: 10 or 48 hours
```

```
> pData(esEset)
```

```
             estrogen time.h
low10-1.cel    absent     10
low10-2.cel    absent     10
high10-1.cel  present     10
high10-2.cel  present     10
low48-1.cel    absent     48
low48-2.cel    absent     48
high48-1.cel  present     48
high48-2.cel  present     48
```

Outlier detection. Before applying linear models to each gene, it may be of interest to investigate the presence of outliers in the data. The single outlier detection method available in factDesign focuses on differences between replicates, thus preserving the independence and normality assumed for the original observations. First, replicate pairs with differences that are significantly larger than expected are identified according to an adjusted F-statistic using the `outlierPair` function. Next, a median absolute deviation filter is applied using `madOutPair` to ensure one of the observations is indeed the single outlier. If no single outlier is detected, `madOutPair` will return `NA`. For example, in Figure 14.4 728_at has a replicate pair with a large difference, but neither observation appears to be outside the range of the other data. On the other hand, 33379_at has one observation that indeed appears to be a single outlier.

```
> library("factDesign")
> op1 <- outlierPair(exprs(esEset)["728_at", ],
+     INDEX = pData(esEset))
> op1
```

```
$test
[1] TRUE

$pval
[1] 0.0143

$whichPair
[1] 7 8
```

```
> madOutPair(exprs(esEset)["728_at", ], op1[[3]])
```

```
[1] NA
```

```
> op2 <- outlierPair(exprs(esEset)["33379_at", ],
+     INDEX = pData(esEset))
> madOutPair(exprs(esEset)["33379_at", ], op2[[3]])
```

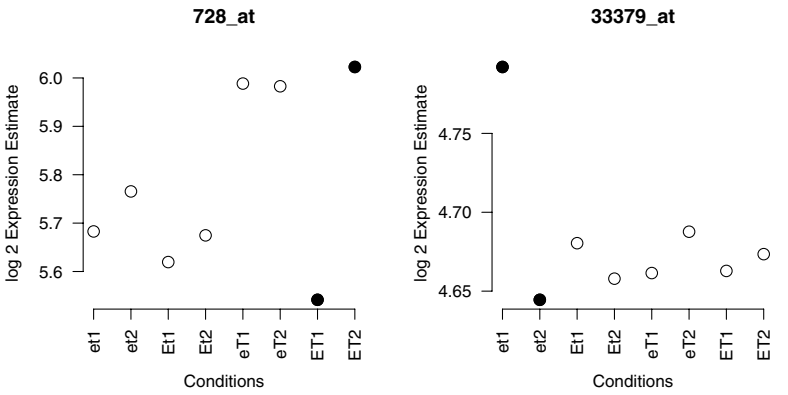

Figure 14.4. Both probesets contain a replicate pair with a larger difference than the other pairs for that probeset, however the single outlier is not obvious for 728_at.

```
[1] 1
```

The user must determine what to do with observations that appear to be single outliers, keeping in mind that removing single outliers assumes that the changes in expression across experimental conditions are small compared to the outlier effects. For probe 33379_at, it could be the second observation that is the outlier if true expression happens to be high at the earlier time in the absence of estrogen. In this application, we choose to leave the single outliers in the data set to preserve the balanced design.

Describing the Linear Model. The 2^2 factorial design of the estrogen experiment makes it a natural fit for linear model analysis. In Equation (14.1), y_{ji} is the observed expression level for gene j in sample i ($i = 1, ..., 8$) with $x_{ESi} = 1$ if estrogen is present and 0 otherwise and $x_{TIMEi} = 1$ if gene expression was measured at 48 hours and 0 otherwise. Using this parameterization, μ_j is the expression level of untreated gene j at 10 hours, β_{ESj} and β_{TIMEj} represent the effects of estrogen and time on the expression level of gene j, respectively, and the interaction term $\beta_{ES:TIMEj}$ quantifies any change in estrogen effect over time for gene j. The error term ϵ_{ji} is assumed to be normally distributed with mean 0 and variance σ_j^2.

$$y_{ji} = \mu_j + \beta_{ESj}x_{ESi} + \beta_{TIMEj}x_{TIMEi} + \beta_{ES:TIMEj}x_{ESi}x_{TIMEi} + \epsilon_{ji} \tag{14.1}$$

We use functions from the limma package to estimate the linear model parameters for every gene using least squares, and call the estimates $\hat{\mu}_j$, $\hat{\beta}_{ESj}$, $\hat{\beta}_{TIMEj}$, and $\hat{\beta}_{ES:TIMEj}$. For gene j, the samples that were not treated with estrogen and were measured at 10 hours will have estimated

expression values of $\hat{\mu}_j$. The estrogen-treated, 10-hour samples will have estimates $\hat{\mu}_j + \hat{\beta}_{ESj}$. The untreated, 48-hour samples will have estimates $\hat{\mu}_j + \hat{\beta}_{TIMEj}$. The estrogen-treated, 48-hour samples will have estimates $\hat{\mu}_j + \hat{\beta}_{ESj} + \hat{\beta}_{TIMEj} + \hat{\beta}_{ES:TIMEj}$. In what follows, we drop the j subscripts for ease of notation, but the linear model parameters are understood to be gene-specific.

```
> pdat <- pData(esEset)
> design <- model.matrix(~factor(estrogen) * factor(time.h),
+     pdat)
> colnames(design) <- c("Intercept", "ES", "T48",
+     "ES:T48")
> fit <- lmFit(esEset, design)
> fit$coefficients[1:3, ]
```

```
          Intercept      ES    T48 ES:T48
1000_at       10.33 -0.3725 -0.122 0.2725
1001_at        5.80  0.1075  0.191 0.0350
1002_f_at      5.66 -0.0676 -0.215 0.1944
```

Suppose we are interested in identifying genes that demonstrate response to estrogen at 10 and/or 48 hours. Genes affected by estrogen at 10 hours will demonstrate a difference in their untreated 10-hour expression levels and their estrogen-treated 10-hour expression levels. Using the linear model parameterization, these genes can be identified as those for which the null hypothesis

$$H_{0,ES10} : \mu = \mu + \beta_{ES} \text{ or } H_{0,ES10} : \beta_{ES} = 0 \tag{14.2}$$

is rejected. Rejection of $H_{0,ES10}$ indicates a difference in the untreated 10-hour and estrogen-treated 10-hour experimental conditions. A similar null hypothesis can be constructed for genes affected by estrogen at 48 hours. We can compare the untreated, 48-hour expression levels to the estrogen-treated 48-hour expression levels by testing the null hypothesis

$$H_{0,ES48} : \mu + \beta_{TIME} = \mu + \beta_{TIME} + \beta_{ES} + \beta_{ES:TIME} \text{ or} \tag{14.3}$$

$$H_{0,ES48} : \beta_{ES} + \beta_{ES:TIME} = 0. \tag{14.4}$$

One way to select genes affected by estrogen at either or both time points is to simultaneously test both contrasts

$$H_{0,ES} : \begin{cases} \beta_{ES} = 0 \\ \beta_{ES} + \beta_{ES:TIME} = 0 \end{cases} \tag{14.5}$$

and then classify the genes according to whether they were affected by estrogen at 10 hours, 48 hours, or both.

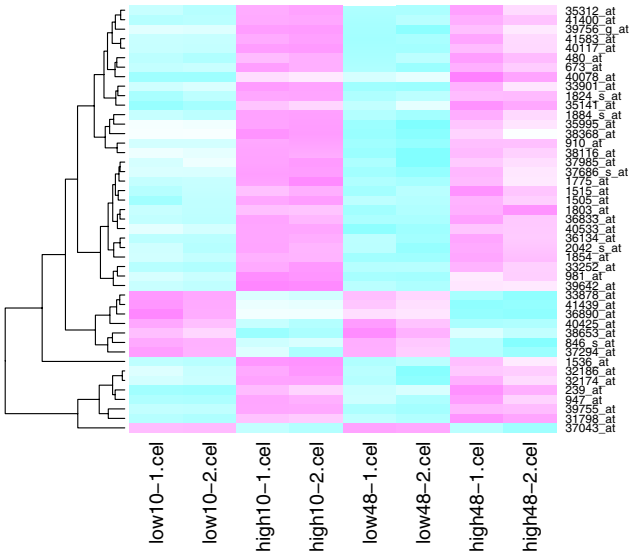

Figure 14.5. Heatmap of expression levels for genes identified as estrogen targets at both 10- and 48-hour time points.

```
> contM <- cbind(es10 = c(0, 1, 0, 0), es48 = c(0,
+     1, 0, 1))
> fitC <- contrasts.fit(fit, contM)
> fitC <- eBayes(fitC)
> esClas <- classifyTestsF(fitC, p = 1e-05)
> print(colSums(abs(esClas)))

es10 es48
  51   83
```

Heatmaps can be a helpful way to visualize the results of linear model analyses for factorial designed experiments. Here we examine three separate heatmaps for genes affected at 10 and 48, only 10, and only 48 hours.

The heatmap in Figure 14.5 helps identify collections of genes that show similar, consistent patterns of up- or down-regulation by estrogen. For these genes, we notice consistent effects for both time points. No genes in this example demonstrate up-regulation at one time point and down-regulation at the other, although such expression behavior could be detected by the contrasts we tested. Further experiments examining the joint behavior of these genes could clarify effects of estrogen on breast cancer cellular pathways that are consistent over time.

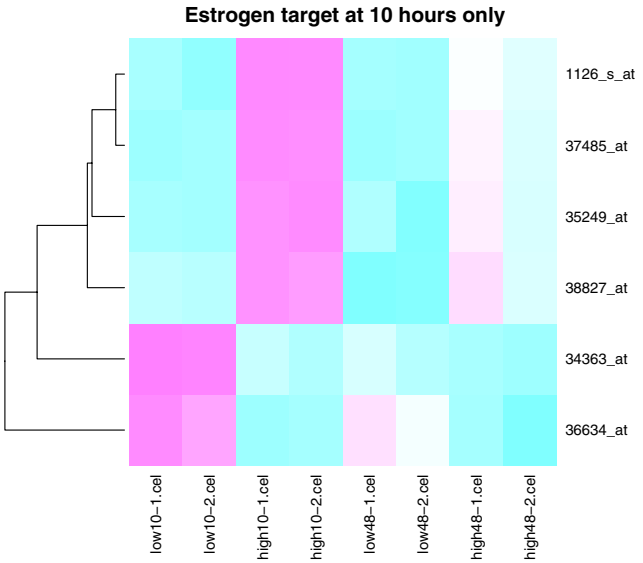

Figure 14.6. Heatmap of expression levels for genes identified as estrogen targets at 10 hours only.

For the 10-hour only target genes, the heatmap in Figure 14.6 identifies two similar clusters. Note that the genes that are up-regulated at 10 hours return to their original expression level at 48 hours, whereas the genes that are down-regulated at 10 hours stay down throughout the course of the experiment.

The heatmap in Figure 14.7 shows genes that were chosen as estrogen targets at 48 hours only and reveals that most of those genes had changes in expression earlier in the experiment. One might conclude that the genes affected at 10 hours comprise the direct targets of estrogen, that is, those genes that are directly stimulated or inhibited by the estrogen-bound ER. The 48-hour targets may be genes further downstream in estrogen-affected pathways. While that is appealing, the time sequence data alone are not strong enough to allow such conclusions.

Multifactor experiments, when designed very carefully with the appropriate biological context and estimable contrasts in mind, can lead to highly informative information regarding the genetic network. As stated previously, the estrogen data set consists of a subset of a larger 2^4 factorial experiment. In additional to estrogen and time, the investigators also exposed the breast cancer cells to cyclohexamide (CX), a translational inhibitor, as well as a drug, here called Z. Translational inhibition by CX

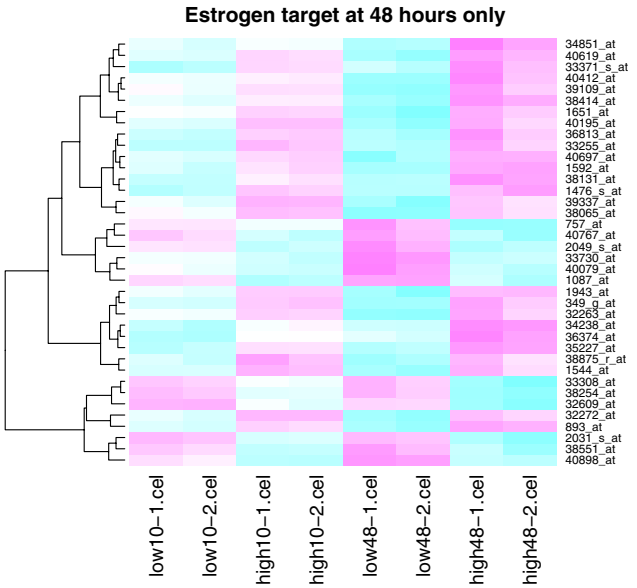

Figure 14.7. Heatmap of expression levels for genes identified as estrogen targets at 48 hours only.

presents a problem for normalization because the presumption that most genes are not differentially expressed is violated. Nevertheless, CX was crucial to the interpretable experimental design as explained in what follows. The full linear model for this factorial experiment consists of all four main effects for CX, ES, Z, and TIME, as well as all possible interactions.

Rather than rely on time sequence alone, CX was a key factor in this experiment for correctly identifying primary and secondary targets. For primary targets, estrogen can cause changes in mRNA levels regardless of the presence of CX. In the presence of CX, however, mRNA from the primary targets cannot be translated into protein, therefore preventing downstream transcriptional changes for the secondary targets. At the ten hour time point, primary estrogen targets were identified by testing

$$H_{0,primary} : \mu + \beta_{CX} = \mu + \beta_{CX} + \beta_{ES} + \beta_{CX:ES} = 0 \text{ or}$$
$$H_{0,primary} : \beta_{ES} + \beta_{CX:ES} = 0 \tag{14.6}$$

as a low p-value for this test of contrast would indicate that the expression level of the gene when exposed only to CX was different than when exposed to both ES and CX. Estrogen targets for which $H_{0,primary}$ (14.6) was not rejected, but $H_{0,ES10}$ (14.2) was rejected, were identified as secondary tar-

gets because they were affected by estrogen, but not in the presence of CX. The fact that CX prevented expression level change due to ES indicated that translation of some other ES target gene's mRNA into protein was required for ES stimulation or repression of the secondary ES target. Similar tests of contrasts were also performed in this experiment to determine which genes were affected by the drug Z, and whether Z executed its action through transcriptional of translational control of the gene expression mechanism.

14.4 Conclusion

In summary, microarrays are used in a wide variety of experimental settings for the detection of differential gene expression. Although the goals and design concerns of these experiments vary, concepts including gene filtering, multiple comparisons adjustment, and gene selection according to the appropriate test statistic apply in general to these experiments. The Bioconductor packages help address these concerns, thereby providing insight into biological pathways and providing a platform for future hypothesis development.