Estrogen 2×2 Factorial Design

Denise Scholtens, Robert Gentleman

Experimental Data

In this vignette, we demonstrate how to use linear models and the package *factDesign* to analyze data from a factorial designed microarray experiment. When careful attention is paid to the biological interpretation of the linear model parameters, multifactor experiments can be particularly useful for disentangling complex biological systems. These methods are also more generally applicable to any microarray experiment to which linear modeling applies. For small experiments, investigators may want to consider moderating variance estimates using the techniques available in packages such as *limma*, *siggenes*, *LPE*, and *EBarrays*.

In this package, an `ExpressionSet` object called `estrogen` contains gene expression levels for 500 genes from Affymetrix HGU95av2 chips for eight samples from a breast cancer cell line. The results of the analysis of the full data set (12,625 probes, 32 samples) are discussed in Scholtens, et al. Analyzing Factorial Designed Microarray Experiments. Journal of Multivariate Analysis. (To appear). The expression estimates were calculated using the rma method after quantile normalization from the *affy* package. The expression values are reported in log base 2 scale as returned by `rma` (Irizarry et al, 2003).

```
> library(Biobase)
> library(affy)
> library(stats)
> library(factDesign)
>
```

The investigators in this experiment were interested in the effect of estrogen on the genes in ER+ breast cancer cells over time. After serum starvation of all eight samples, they exposed four samples to estrogen, and then measured mRNA transcript abundance after 10 hours for two samples and 48 hours for the other two. They left the remaining four samples untreated, and measured mRNA transcript abundance at 10 hours for two samples, and 48 hours for the other two. Since there are two factors in this experiment (*estrogen* and *time*), each at two levels, (*estrogen*: *absent* or *present*,*time*: *10 hours* or *48 hours*), this experiment is said to have a 2×2 factorial design. Table 1 shows the correspondence of the sample names in `estrogen` with the experimental conditions.

```
> data(estrogen)
> estrogen

ExpressionSet (storageMode: lockedEnvironment)
assayData: 500 features, 8 samples
```

Table 1: Experimental Conditions for `.cel` Files

time	estrogen	
	absent	present
10 hours	et1	Et1
	et2	Et2
48 hours	eT1	ET1
	eT2	ET2

```
  element names: exprs, se.exprs
protocolData: none
phenoData
  sampleNames: et1.CEL et2.CEL ... ET2.CEL (8 total)
  varLabels: ES TIME
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
Annotation: hgu95av2
```

```
> pData(estrogen)
```

```
        ES TIME
et1.CEL  A  10h
et2.CEL  A  10h
Et1.CEL  P  10h
Et2.CEL  P  10h
eT1.CEL  A  48h
eT2.CEL  A  48h
ET1.CEL  P  48h
ET2.CEL  P  48h
```

```
>
```

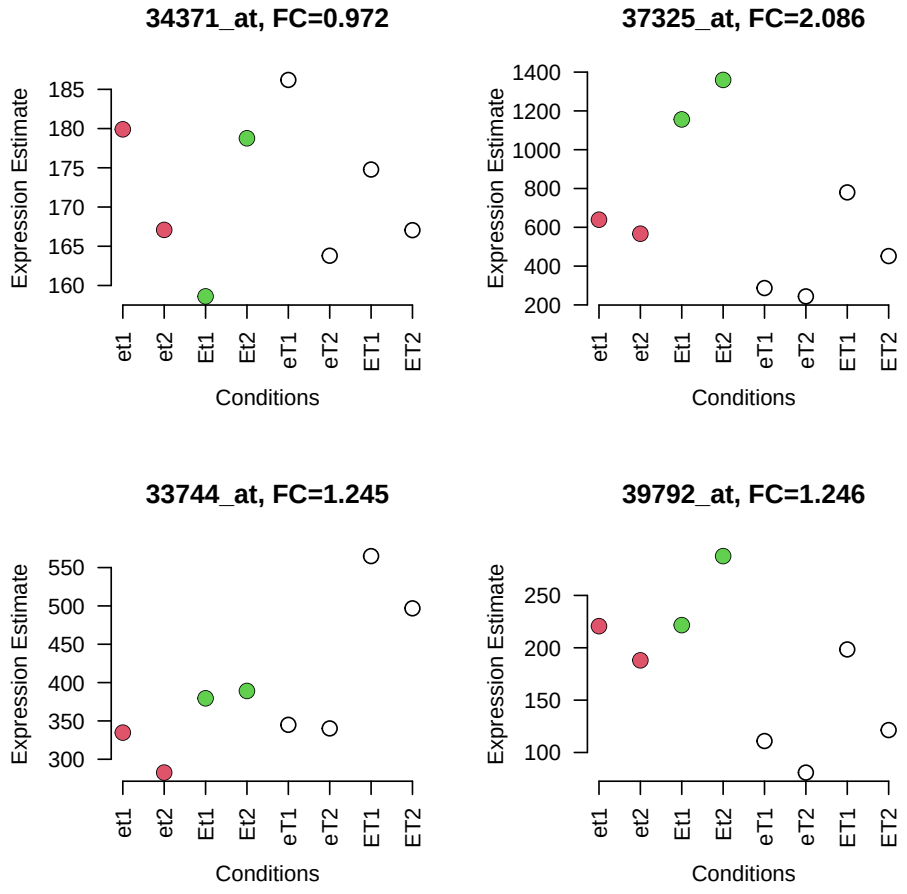
Analysis Using Fold Change Criteria

A simple method for finding estrogen-affected genes would be to form a ratio of the mean expression levels of the estrogen-treated samples to the mean of the expression levels for the untreated samples. Suppose we consider only the 10-hour time point, calculate fold change (FC) values for the estrogen-treated vs. untreated samples, and select genes for which we observe FC>2. In the plots below, absence/presence of estrogen is represented by `e/E` and the 10/48 hour time point is represented by `t/T` on the horizontal axis. The proposed FC criteria at 10 hours would compare the mean of the green dots to the mean of the red dots.

If we used a FC$>$ 2 criteria to identify ES-affected genes in the `estrogen` data set, we would successfully eliminate genes like 34371_at and select genes like 37325_at; however, we could miss

several interesting genes. For example, 33744_at has a lower fold change value of 1.245, but the replicates are very consistent, leading us to believe that this smaller effect might be real. We would want to distinguish this from other genes like 39792_at which has a similar fold change value of 1.246, but quite variable observations.

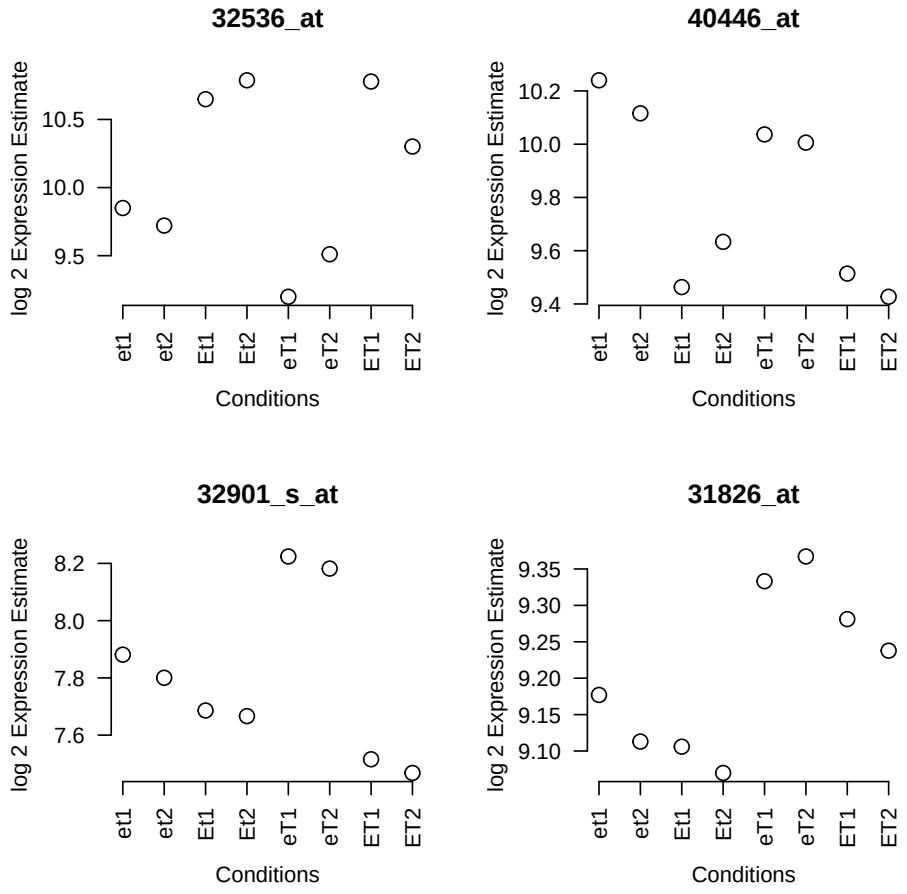
```
> par(mfrow=c(2,2))
> par(las=2)
> for (i in c("34371_at","37325_at","33744_at","39792_at")){
+         expvals <- 2^exprs(estrogen)[i,]
+         plot(expvals,axes=F,cex=1.5,
+                 xlab="Conditions",ylab="Expression Estimate")
+         points(1:2,expvals[1:2],pch=16,cex=1.5,col=2)
+         points(3:4,expvals[3:4],pch=16,cex=1.5,col=3)
+         axis(1,at=1:8,labels=c("et1","et2","Et1","Et2","eT1","eT2","ET1","ET2"),
+         axis(2)
+         FC <- round(mean(expvals[3:4])/mean(expvals[1:2]),3)
+         title(paste(i,", FC=",FC,sep=""))
+ }
>
```



We would like to find genes with consistent expression estimates between replicate samples that are either up- or down-regulated by estrogen, for example 32536_at and 40446_at. We would also like to find genes like gene 32901_at for which the magnitude of the effect of estrogen changes over time. Furthermore, we would like to exclude genes like 31826_at that demonstrate change primarily over time, and not necessarily due to estrogen.

Selecting genes according to fold change estimates alone does not take advantage of the measure of variability in gene expression offered by the replicate sampels. Furthermore, we cannot attach statistical significance (i.e., a p-value) to the fold change estimates computed in this manner. It is also difficult to quantify any change in estrogen effect over time. Classical statistical linear modeling with thoughtful biological interpretation of the parameters offers a natural paradigm for the analysis of factorial designed microarray experiments.

```
> par(mfrow=c(2,2))
> par(las=2)
> for (i in c("32536_at","40446_at","32901_s_at","31826_at")){
+         expvals <- exprs(estrogen)[i,]
+         plot(expvals,axes=F,cex=1.5,
+                 xlab="Conditions",ylab="log 2 Expression Estimate")
+         axis(1,at=1:8,labels=c("et1","et2","Et1","Et2","eT1","eT2","ET1","ET2"),
+         axis(2)
+         title(i)
+ }
>
```



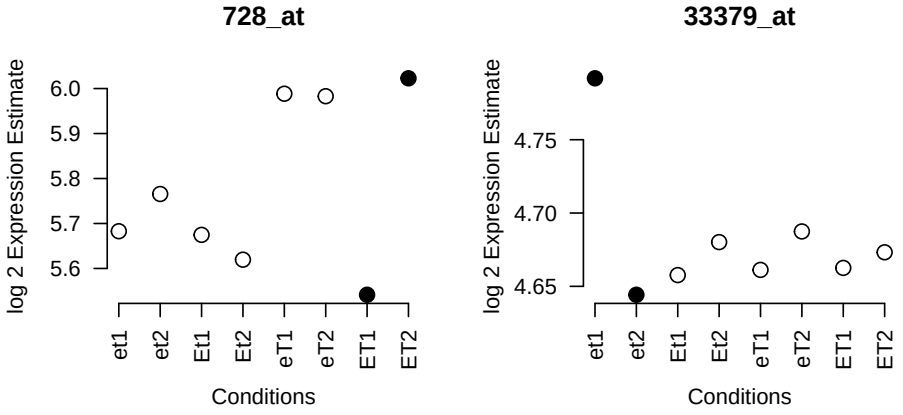
Removing Outliers

Before defining the linear model for this particular experiment, we may want to remove observations that might be single outliers in the data set. The test we used is based on the differences between replicates and is appropriate for small factorial experimental designs. First, we identify replicate pairs with differences that are significantly larger than expected, and then we can apply a median absolute deviation filter to make sure one of the observations is indeed the single outlier. For example, 728_at has a replicate pair with a large difference, but we wouldn't want to label either observation as the single outlier. 33379_at has one observation that indeed appears to be a single outlier.

Removing single outliers from small factorial designed experiments does assume that the changes in expression across experimental conditions are small compared to the outlier effects. For probe 33379_at, it could very well be the second observation which is the outlier if true expression happens to be higher at the earlier time in the presence of estrogen. Users should consider whether or not single outlier elimination is appropriate in their particular experimental setting. Here we have commented out the code that could be used to replace single outliers with "NA" values.

```
> op1 <- outlierPair(exprs(estrogen)["728_at",],INDEX=pData(estrogen),p=.05)
> print(op1)
```

```
$test
[1] TRUE

$pval
[1] 0.01432178

$whichPair
[1] 7 8

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

[1] NA

> par(mfrow=c(2,2))
> par(las=2)
> for (i in c("728_at","33379_at")){
+         expvals <- exprs(estrogen)[i,]
+         plot(expvals,axes=F,cex=1.5,
+                 xlab="Conditions",ylab="log 2 Expression Estimate")
+         if(i=="728_at") points(7:8,expvals[7:8],pch=16,cex=1.5)
+         if(i=="33379_at") points(1:2,expvals[1:2],pch=16,cex=1.5)
+         axis(1,at=1:8,labels=c("et1","et2","Et1","Et2","eT1","eT2","ET1","ET2"),
+         axis(2)
+         title(i)
+ }
>
> #for (j in 1:500){
> #        op <- outlierPair(exprs(estrogen)[j,],INDEX=pData(estrogen),p=.05)
> #        if(op[[1]]){
> #                so <- madOutPair(exprs(estrogen)[j,],op[[3]])
> #                if(!is.na(so)) exprs(estrogen)[j,so] <- NA
> #        }
> #}
>
>
```



Describing the Linear Model

The 2×2 factorial design of this experiment allows us to use a statistical linear model to measure the effects of estrogen and time on gene expression. In equation (1), $y_{full,ij}$ is the observed expression level for gene i in sample j ($j = 1, ..., 8$). $x_{ESj} = 1$ if estrogen is present and 0 otherwise; $x_{TIMEj} = 1$ if gene expression was measured at 48 hours and 0 otherwise. μ_i is the expression level of untreated gene i at 10 hours. β_{ESi} and β_{TIMEi} represent the effects of estrogen and time on the expression level of gene i, respectively. $\beta_{ES:TIMEi}$ is called an interaction term for gene i; this allows us to quantify any change in estrogen effect over time for probes like `1700_at`. ϵ_{ij} represents random error for gene i and sample j, and is assumed to be independent for each gene and sample, and normally distributed with mean 0 and variance σ_i^2. The biologically independent replicates of the experimental conditions in this study allow us to estimate σ_i^2.

$$y_{ij} = \mu_i + \beta_{ESi}x_{ESj} + \beta_{TIMEi}x_{TIMEj} + \beta_{ES:TIMEi}x_{ESj}x_{TIMEj} + \epsilon_{ij} \quad (1)$$

To proceed with the analysis, we estimate the β parameters for every gene using least squares, and call the estimates $\hat{\beta}_{ESi}$, $\hat{\beta}_{TIMEi}$, and $\hat{\beta}_{ES:TIMEi}$. For gene i, the samples that were not treated with estrogen and were measured at 10 hours will have estimated expression values of $\hat{\mu}_i$. The estrogen-treated,

10-hour samples will have estimates $\hat{\mu}_i + \hat{\beta}_{ESi}$. The untreated, 48-hour samples will have estimates $\hat{\mu}_i + \hat{\beta}_{TIMEi}$. The estrogen-treated, 48-hour samples will have estimates $\hat{\mu}_i + \hat{\beta}_{ESi} + \hat{\beta}_{TIMEi} + \hat{\beta}_{ES:TIMEi}$.

We will also form a reduced model with only an effect for time (2), and use this to decide if a model including estrogen is appropriate for the gene of interest.

$$y_{ij} = \mu_i + \beta_{TIMEi} x_{TIMEj} + \epsilon_i \tag{2}$$

```
> lm.full <- function(y) lm(y ~ ES + TIME + ES:TIME)
> lm.time <- function(y) lm(y ~ TIME)
> lm.f <- esApply(estrogen, 1, lm.full)
> lm.t <- esApply(estrogen, 1, lm.time)
> lm.f[[1]]

Call:
lm(formula = y ~ ES + TIME + ES:TIME)

Coefficients:
(Intercept)          ESP      TIME48h  ESP:TIME48h
    4.81164     -0.22762      0.01055      0.03839

> lm.t[[1]]

Call:
lm(formula = y ~ TIME)

Coefficients:
(Intercept)      TIME48h
    4.69783      0.02974

>
```

Selecting Genes of Interest using the Linear Model

We are only interested in genes which are affected by estrogen. One way to select such genes is to compare the full linear model (`lm.f`) to the linear model consisting of only a term for time (`lm.t`) using an ANOVA F-test. If the full model `lm.f` fits better than the reduced model `lm.t`, then we know the gene must be affected by estrogen.

Since we have so many genes to consider, multiple comparisons is an obvious problem. The R package *multtest* contains many functions that are suitable for multiple comparisons adjustment for microarrays. Here, the p-values from the ANOVA F-tests are adjusted according to the Benjamini and Hochberg (1995) False Discovery Rate method with an FDR of .15.

```
> Fpvals <- rep(0,length(lm.f))
> for(i in 1:length(lm.f)){
+   Fpvals[i]<-anova(lm.t[[i]],lm.f[[i]])$P[2]
```

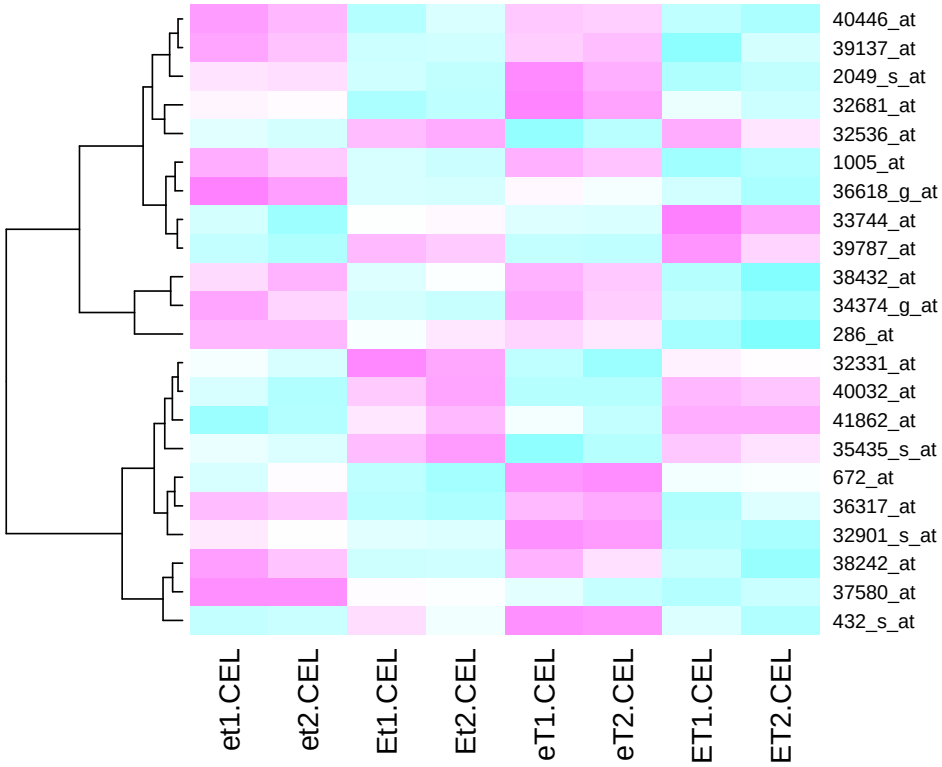
```
+ }
> library(multtest)
> procs <- c("BH")
> F.res <- mt.rawp2adjp(Fpvals,procs)
> F.adjps <- F.res$adjp[order(F.res$index),]
> Fsub <- which(F.adjps[,"BH"]<.15)
> estrogen.Fsub <- estrogen[Fsub]
> lm.f.Fsub <- lm.f[Fsub]
> estrogen.Fsub

ExpressionSet (storageMode: lockedEnvironment)
assayData: 28 features, 8 samples
  element names: exprs, se.exprs
protocolData: none
phenoData
  sampleNames: et1.CEL et2.CEL ... ET2.CEL (8 total)
  varLabels: ES TIME
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
Annotation: hgu95av2

>
>
```

Suppose we want to identify genes that are affected by estrogen at 10 hours. In our linear model, this corresponds to testing a null hypothesis $H_{0ES} : \beta_{ES} = 0$, and if the hypothesis rejected, concluding that the gene has a main estrogen effect.

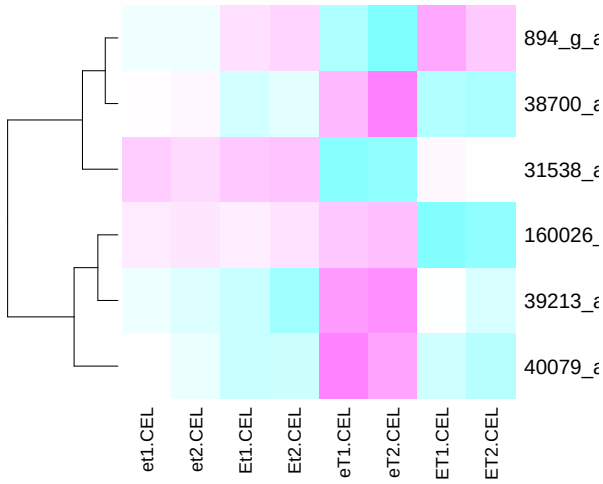
```
> betaNames <- names(coef(lm.f[[1]]))
> lambda <- par2lambda(betaNames,c("ESP"),c(1))
> mainES <- function(x) contrastTest(x,lambda,p=0.05)[[1]]
> mainESgenes <- sapply(lm.f.Fsub,FUN=mainES)
> sum(mainESgenes=="REJECT")

[1] 22

>
```

Heatmaps can be a useful way to visualize genes that are selected according to a certain criteria. In the first heatmap that follows, we see genes for which the null hypothesis H_{0ES} was rejected at a 0.05 significance level. In the second heatmap, we see the genes for which the main estrogen effect was not statistically significant; it appears that estrogen affected these genes only after 48 hours.

```
> heatmap(exprs(estrogen.Fsub)[mainESgenes=="REJECT",],Colv=NA,col=cm.colors(256),
>
```




```
> heatmap(exprs(estrogen.Fsub)[mainESgenes=="FAIL TO REJECT",],Colv=NA,col=cm.colo
>
```



Selecting genes according to p-value can produce some possibly misleading results. For example, 32901_s_at had a main ES effect with p-value for β_{ES} less than 0.01, but the estimate of fold change supression at 10 hours is only .8922. While this small effect is statistically significant, it may not be biologically interesting. Selecting genes with significant contrast test p-values as well as fold change values above a certain threshold can give a good approximation to more sophisticated variance moderating analyses.

```
> lambdaNum <- par2lambda(betaNames,list(c("(Intercept)","ESP")),list(c(1,1)))
> lambdaDenom <-  par2lambda(betaNames,list(c("(Intercept)")),list(c(1)))
> FCval <- findFC(lm.f.Fsub[["32901_s_at"]],lambdaNum,lambdaDenom,logbase=2)
> print(FCval)

          [,1]
[1,] 0.8922424

> FCvals <- lapply(lm.f.Fsub,FUN=findFC,lambdaNum,lambdaDenom,logbase=2)
> largeFC <- unlist(FCvals>1.4 | FCvals<.7)
> estrogen.Fsub.FC <- estrogen.Fsub[largeFC & mainESgenes=="REJECT"]
> heatmap(exprs(estrogen.Fsub.FC),Colv=NA,col=cm.colors(256))
>
```

Now suppose we want to find genes that are affected by estrogen after both 10 and 48 hours. By testing for the main estrogen effect, we have already found genes with an estrogen effect at 10 hours. To select genes with an estrogen effect at 48 hours, we want to compare the gene expression levels of the untreated samples that were measured at 48 hours with the estrogen-treated samples at 48 hours. In terms of our linear model, for each gene, we want to test the null hypothesis $H_{0ES,TIME}$ in (3).

$$H_{0ES,TIME} : \mu + \beta_{TIME} = \mu + \beta_{ES} + \beta_{TIME} + \beta_{ES:TIME} \tag{3}$$

Testing the null hypothesis $H_{0ES,TIME}$ is equivalent to testing the linear contrast $H_{0ES,TIME*}$ in (4).

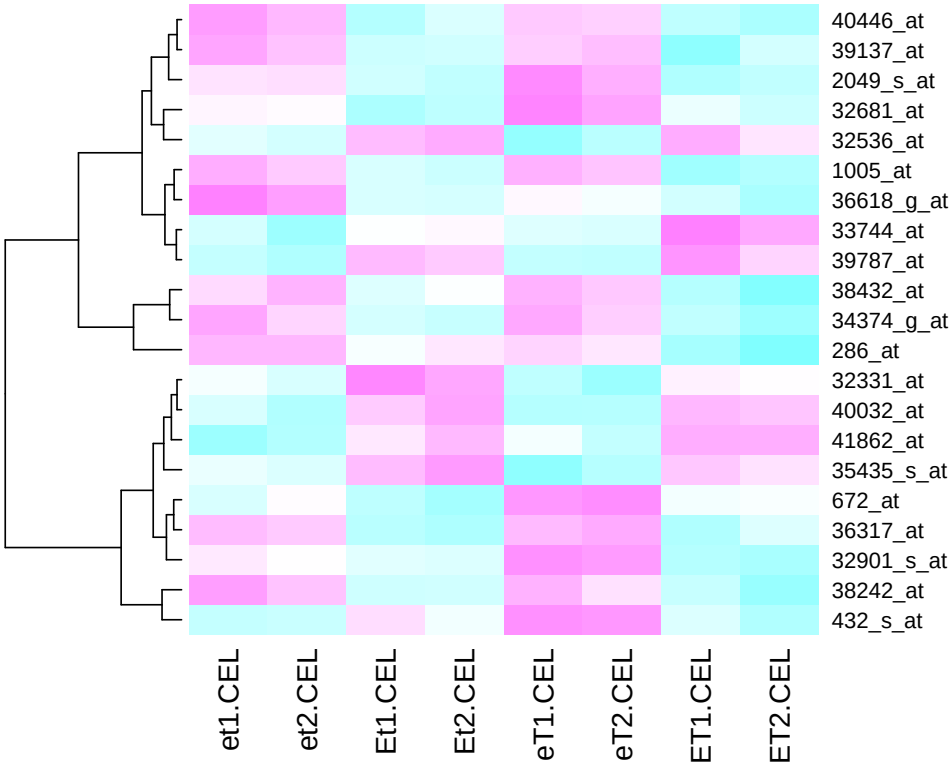
$$H_{0ES,TIME*} : \beta_{ES} + \beta_{ES:TIME} = 0 \quad (4)$$

The technique for testing this linear contrast follows from straightforward linear model theory. The `par2lambda` function helps set up the appropriate matrix for testing sets of linear contrasts.

```
> lambdaEST <- par2lambda(betaNames,list(c("ESP","ESP:TIME48h")),list(c(1,1)))
> ESTcontrast <- function(x) contrastTest(x,lambdaEST,p=.10)[[1]]
> ESTgenes <- sapply(lm.f.Fsub,FUN=ESTcontrast)
> sum(ESTgenes=="REJECT")

[1] 27

>
```

Again, we can use a heatmap to look at genes for which we rejected both H_{0ES} and $H_{0ES,TIME*}$.

```
> heatmap(exprs(estrogen.Fsub)[mainESgenes=="REJECT" & ESTgenes=="REJECT",],Colv=
>
```



After genes are selected according to contrast tests of interest, the annotation information available in other Bioconductor packages allows for more in-depth research on specific genes.

Using linear models for factorial designed microarray experiments enables investigators to extend analyses beyond basic gene filtering according to fold change. Genes can be selected in a high-throughput manner with biologically interpretable parameters and quantifiable measures of confidence. This lab investigated the effects of estrogen on breast cancer cells, but the principles behind this specific example are applicable to any carefully designed microarray study.