# More on differential expression with the limma package

## Author: Georgios Asimomitis

## Introduction

The purpose of this exercise is to understand a few more details of a standard 'limma' differential expression (DE) analysis. In particular, we explore: 1. the combination of design matrices and contrast matrices to answer DE questions-of-interest 2. some of the preprocessing steps (and the concepts leading to them) for Affymetrix microarray data.

Initially we load the necessary libraries and unzip the file "affy_estrogen.zip"

```
library("limma")
library("affy")
library("preprocessCore")
unzip("affy_estrogen.zip")
ddir <- "affy_estrogen"
dir(ddir)
```

```
## [1] "high10-1.cel" "high10-2.cel" "high48-1.cel" "high48-2.cel"
## [5] "low10-1.cel"  "low10-2.cel"  "low48-1.cel"  "low48-2.cel"
## [9] "targets.txt"
```

The details of the experiment are stored in the machine-readable table called "targets.txt". This is our metadata. In the first place we read in the "targets.txt" which contains 8 data files that include the gene differential expression according to time and the presence of estrogen. Then we read in the Affymetrix data and process it with the method RMA (robust multichip analysis) which converts the AffyBatch "abatch" into the ExpressionSet "eset". This expressionSet includes the information for 12625 genes which are expressed over the 8 samples: low10-1.cel, low10-2.cel, high10-1.cel, high10-2.cel, low48-1.cel, low48-2.cel, high48-1.cel, high48-2.cel. "High" and "low" denote the presence or absence of estrogen correspondingly and "10", "48" display the time in hours. Each value of the expressionSet presents the level of gene expression of each gene in the corresponding sample. Our target is to find how many and which genes are differentially expressed in accordance to the different conditions that the samples represent.

```
# preprocess affymetrix data
targets <- readTargets("targets.txt", path=ddir)
targets
```

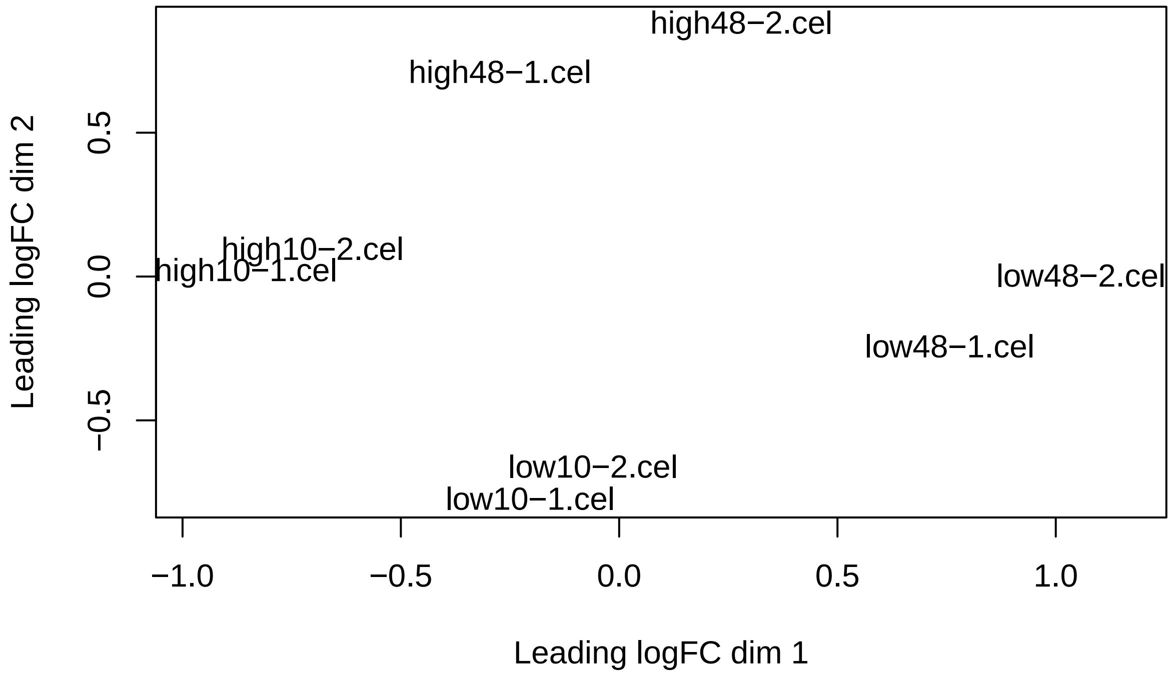
```
##       filename estrogen time.h
## 1  low10-1.cel   absent     10
## 2  low10-2.cel   absent     10
## 3 high10-1.cel  present     10
## 4 high10-2.cel  present     10
## 5  low48-1.cel   absent     48
## 6  low48-2.cel   absent     48
## 7 high48-1.cel  present     48
## 8 high48-2.cel  present     48
```

```r
abatch <- ReadAffy(filenames=targets$filename,
                   celfile.path=ddir)
eset <- rma(abatch)  # bg correct, normalize, summarize
```

```
## Background correcting
## Normalizing
## Calculating Expression
```

In order to have an overall look of our large dataset, we use the multidimensional scaling (MDS) plot to visualize the relations between our samples. In this case, distances on the plot approximate the typical log2 fold changes.

```r
plotMDS( exprs(eset) )  # MDS plot
```



## Design Matrix

An essential step in order to run the standard limma pipeline for differential expression is the construction of the design matrix, which in combination with the optional contrast matrix, models our experiment.

The following design matrix has 4 columns and 8 rows. Each row corresponds to each sample and each of the columns corresponds to one of the 4 different conditions in which we test gene expression; the first column "absent10" specifies the samples (1,2) in which time equals to 10 hours and estrogen was almost absent, the second column "absent48" specifies the samples (5,6) in which time equals to 48 hours and estrogen was absent as well, the third column "present10" specifies the samples (3,4) in which time equals to 10 hours and estrogen was present and the fourth column "present48" specifies the samples (7,8) in which time equals to 48 hours and estrogen was present as well. The value of 1 in the design matrix represents the sample participation in the corresponding column.

The form of the design matrix is in accordance with the parameters that need to be estimated. In our model we define the parameter vector b=transpose([b1 b2 b3 b4]), where b1 denotes the condition "absent10", b2 denotes the condition "absent48", b3 denotes the condition "present10" and b4 denotes the condition "present48".

In the code below, the metadata is encoded into a factor variable that is used for creating the design matrix.

```
# do the limma modeling
f <- paste(targets$estrogen,targets$time.h,sep="")
f <- factor(f)

# create design matrix
design <- model.matrix(~0+f)
colnames(design) <- levels(f)
design
```

```
##   absent10 absent48 present10 present48
## 1        1        0         0         0
## 2        1        0         0         0
## 3        0        0         1         0
## 4        0        0         1         0
## 5        0        1         0         0
## 6        0        1         0         0
## 7        0        0         0         1
## 8        0        0         0         1
## attr(,"assign")
## [1] 1 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$f
## [1] "contr.treatment"
```

We can now fit the linear model using LmFit which takes as input the expression set and the design matrix.

```
fit <- lmFit(eset, design)
```

# Contrast Matrix

In order to draw conclusions about the differential expression across the different experimental condtions we need to interpret and process the parameters of our design matrix in a useful way. Therefore, we define a contrast matrix, which defines the specific comparisons in which we are interested by forming the substractions between the appropriate parameters. In particular, we define 3 contrast variables: E10 which represents the difference between present10 and absent10, E48 which represents the difference between present48 and absent48 and Time which represents the difference between absent48 and absent10. E10 contrast variable refers to the case in which genes are differentially expressed in the presence of estrogen in the time scale of 10 hours and are not differentially expressed in the absent10 condition. Similarly, E10 contrast variable describes the case in which genes are differentially expressed in present48 and not in absent48 and Time refers to the ones that in the absense of estrogen are differentially expressed in the scale of 48 hours and not in 10h.

Since the contrast matrix constructs 3 contrast variables out of 4 parameters, it has 3 columns and 4 rows. The Value of -1 denotes a parameter that is subtracted in the corresponding contrast variable, a value of 1 shows a parameter out of which a value is subtracted and zero denotes that a parameter does not participate in the formation of the contrast variable.

The contrast matrix is constructed by using the makeContrasts() accessory function.

```
cont.matrix <- makeContrasts(E10="present10-absent10",
                             E48="present48-absent48",
                             Time="absent48-absent10",levels=design)
cont.matrix
```

```
##            Contrasts
## Levels      E10 E48 Time
##   absent10   -1   0   -1
##   absent48    0  -1    1
##   present10   1   0    0
##   present48   0   1    0
```

## Model, Coefficients, Top Genes

Now, the contrasts can be fit and the moderation of the variance parameters can be performed. Given the linear model fit, contrasts.fit function computes the estimated coefficients and standard errors for the given set of contrasts. The output is an object of the same class as fit, (MArrayLM) that includes a numeric matrix containing the estimated coefficients for each contrast.

Then we use eBayes built in R function which takes as input the linear model and computes moderated t-statistics, moderated F-statistics, and log-odds of differential expression by empirical Bayes moderation of the standard errors towards a common value.

```
fit2  <- contrasts.fit(fit, cont.matrix)
fit2  <- eBayes(fit2)
fit2
```

```
## An object of class "MArrayLM"
## $coefficients
##            Contrasts
##                     E10         E48       Time
##   100_g_at  -0.24686537 -0.07791790 -0.1110524
##   1000_at   -0.37251645 -0.09999976 -0.1220724
##   1001_at    0.10748492  0.14242137  0.1914043
##   1002_f_at -0.06760315  0.12681750 -0.2149139
##   1003_s_at  0.04060842  0.08145908  0.1367273
## 12620 more rows ...
##
## $rank
## [1] 4
##
## $assign
## [1] 1 1 1 1
##
## $qr
## $qr
##     absent10   absent48  present10  present48
## 1 -1.4142136  0.0000000  0.0000000  0.0000000
## 2  0.7071068 -1.4142136  0.0000000  0.0000000
## 3  0.0000000  0.0000000 -1.4142136  0.0000000
## 4  0.0000000  0.0000000  0.7071068 -1.4142136
## 5  0.0000000  0.7071068  0.0000000  0.0000000
## 6  0.0000000  0.7071068  0.0000000  0.0000000
## 7  0.0000000  0.0000000  0.0000000  0.7071068
## 8  0.0000000  0.0000000  0.0000000  0.7071068
## attr(,"assign")
## [1] 1 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$f
## [1] "contr.treatment"
##
##
## $qraux
## [1] 1.707107 1.000000 1.707107 1.000000
##
## $pivot
## [1] 1 2 3 4
##
## $tol
## [1] 1e-07
##
## $rank
## [1] 4
##
##
## $df.residual
## [1] 4 4 4 4 4
## 12620 more elements ...
##
## $sigma
##   100_g_at    1000_at    1001_at  1002_f_at  1003_s_at
## 0.07649887 0.21776253 0.12832927 0.11769183 0.15544260
## 12620 more elements ...
##
## $cov.coefficients
##          Contrasts
## Contrasts E10  E48 Time
##      E10  1.0  0.0  0.5
##      E48  0.0  1.0 -0.5
##      Time 0.5 -0.5  1.0
##
## $stdev.unscaled
##            Contrasts
##             E10 E48 Time
##   100_g_at    1   1    1
##   1000_at     1   1    1
##   1001_at     1   1    1
##   1002_f_at   1   1    1
##   1003_s_at   1   1    1
## 12620 more rows ...
```

```
##
## $Amean
##  100_g_at   1000_at   1001_at 1002_f_at 1003_s_at
##  9.555474 10.147100  5.957489  5.564548  7.994831
## 12620 more elements ...
##
## $method
## [1] "ls"
##
## $design
##   absent10 absent48 present10 present48
## 1        1        0         0         0
## 2        1        0         0         0
## 3        0        0         1         0
## 4        0        0         1         0
## 5        0        1         0         0
## 6        0        1         0         0
## 7        0        0         0         1
## 8        0        0         0         1
## attr(,"assign")
## [1] 1 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$f
## [1] "contr.treatment"
##
##
## $contrasts
##            Contrasts
## Levels      E10 E48 Time
##   absent10   -1   0   -1
##   absent48    0  -1    1
##   present10   1   0    0
##   present48   0   1    0
##
## $df.prior
## [1] 4.479981
##
## $s2.prior
## [1] 0.02199399
##
## $var.prior
## [1] 65.01775 93.87792 52.61593
##
## $proportion
## [1] 0.01
##
## $s2.post
##   100_g_at    1000_at    1001_at  1002_f_at  1003_s_at
## 0.01437986 0.03398766 0.01938757 0.01815312 0.02301683
## 12620 more elements ...
##
## $t
##            Contrasts
##                    E10        E48       Time
##   100_g_at  -2.0586514 -0.6497704 -0.9260848
##   1000_at   -2.0206207 -0.5424233 -0.6621509
##   1001_at    0.7719440  1.0228535  1.3746428
##   1002_f_at -0.5017545  0.9412468 -1.5951037
##   1003_s_at  0.2676661  0.5369289  0.9012233
## 12620 more rows ...
##
## $df.total
## [1] 8.479981 8.479981 8.479981 8.479981 8.479981
## 12620 more elements ...
##
## $p.value
##            Contrasts
##                     E10       E48      Time
##   100_g_at  0.07153470 0.5330438 0.3800031
##   1000_at   0.07597743 0.6014878 0.5254577
##   1001_at   0.46111710 0.3346688 0.2044708
##   1002_f_at 0.62860708 0.3726157 0.1472181
##   1003_s_at 0.79535224 0.6051150 0.3923460
## 12620 more rows ...
##
## $lods
##            Contrasts
##                   E10       E48      Time
##   100_g_at  -4.804654 -6.643594 -6.138313
##   1000_at   -4.861525 -6.711478 -6.351663
##   1001_at   -6.373222 -6.326130 -5.652146
##   1002_f_at -6.553538 -6.405619 -5.368735
##   1003_s_at -6.650810 -6.714647 -6.160958
## 12620 more rows ...
##
## $F
## [1] 1.5861664 1.4629910 2.0475707 0.8880226 0.8353092
## 12620 more elements ...
##
## $F.p.value
## [1] 0.2634847 0.2922784 0.1817192 0.4855166 0.5094259
## 12620 more elements ...
```

```
class(fit2)
```

```
## [1] "MArrayLM"
## attr(,"package")
## [1] "limma"
```

```
names(fit2)
```

```
##  [1] "coefficients"     "rank"             "assign"
##  [4] "qr"               "df.residual"      "sigma"
##  [7] "cov.coefficients" "stdev.unscaled"   "Amean"
## [10] "method"           "design"           "contrasts"
## [13] "df.prior"         "s2.prior"         "var.prior"
## [16] "proportion"       "s2.post"          "t"
## [19] "df.total"         "p.value"          "lods"
## [22] "F"                "F.p.value"
```

Since our model has fit three contrasts variables, the matrix of coefficients contains 3 columns.

```
dim(fit2$coefficients)
```

```
## [1] 12625     3
```

```
colnames(fit2$coefficients)
```

```
## [1] "E10"  "E48"  "Time"
```

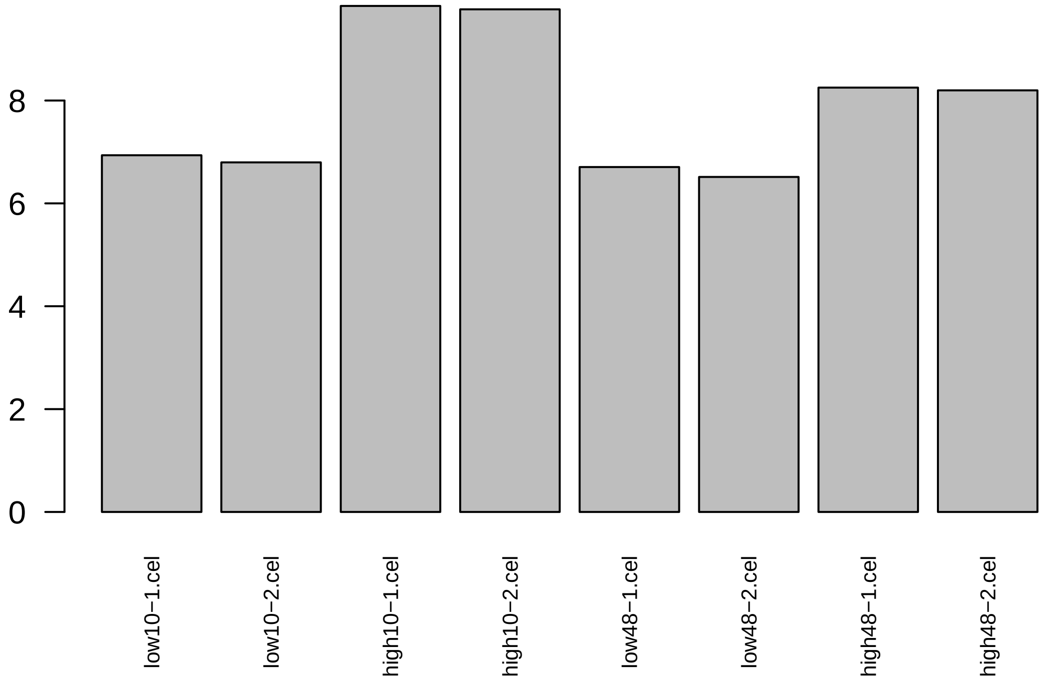
Coef = 1 corresponds to the first contrast variable E10, coef = 2 refers to the second contrast variable E48 and coef = 3 refers to the contrast variable Time. In order to extract a table of the top-ranked genes from a linear model fit given a specific contrast variable we make use of the built in R function topTable.

```
topTable(fit2,coef=1)
```

```
##               logFC   AveExpr        t      P.Value    adj.P.Val        B
## 39642_at  2.939428  7.876515 23.71715 4.741579e-09 3.128295e-05 9.966810
## 910_at    3.113733  9.660238 23.59225 4.955715e-09 3.128295e-05 9.942522
## 31798_at  2.800195 12.115778 16.38509 1.025747e-07 3.511070e-04 7.977290
## 41400_at  2.381040 10.041553 16.22463 1.112418e-07 3.511070e-04 7.916921
## 40117_at  2.555282  9.676557 15.68070 1.472942e-07 3.576234e-04 7.705093
## 1854_at   2.507616  8.532099 15.15848 1.945518e-07 3.576234e-04 7.490766
## 39755_at  1.679331 12.131839 15.06365 2.048314e-07 3.576234e-04 7.450643
## 1824_s_at 1.914637  9.238870 14.87915 2.266129e-07 3.576234e-04 7.371475
## 1126_s_at 1.782825  6.879918 13.83040 4.119252e-07 5.778395e-04 6.892307
## 1536_at   2.662258  5.937222 13.26247 5.795111e-07 7.316327e-04 6.610486
```

For the first contrast variable E10, the top ranked gene is “39642_at”. This gene has the highest computed t-value in comparison with the others for this contrast variable. In other words the difference in the mean values of gene expression between the present10 and absent10 samples divided by its standard error, (which in moderated t-statistics has been moderated across genes effectively borrowing information from the ensemble of genes to aid with inference about each individual gene) has the highest value in this particular gene. By using barplot we can indeed observe the difference in the levels of gene expression between the samples “high10-1.cel” and “low10-1.cel” as well as “high10-2.cel” and “low10-2.cel”.

```
barplot( exprs(eset)["39642_at",], las=2, cex.names=.7 )  # top gene
```

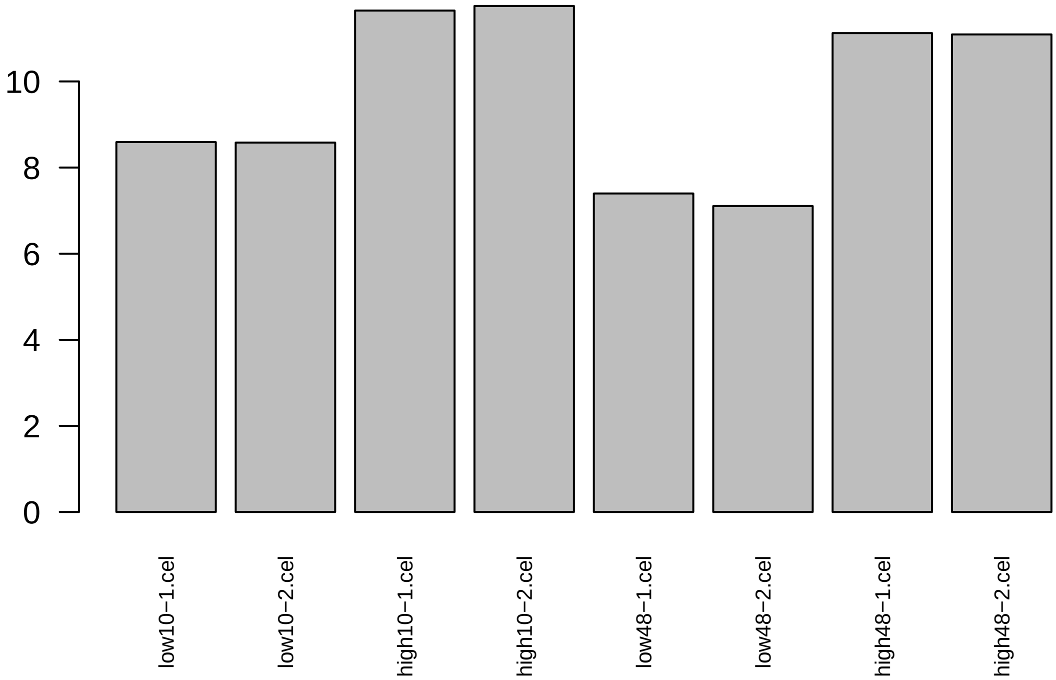


As far as E48 is concerned, we observe below that the top gene is “910_at”, which happens also to be the second ranked gene for E10. Out of this we can infer its ability to be differentially expressed in the conditions where estrogen is present regardless of time. This is depicted also in the following barplot where high levels of gene expression are presented in the samples “high10-1.cel”, “high10-2.cel”, “high48-1.cel”, and “high48-2.cel”. However, E48 refers only to the comparison between “high48-1.cel” and “low48-1.cel” as well as “high48-2.cel” and “low48-2.cel”.

```
topTable(fit2,coef=2)
```

```
##              logFC   AveExpr         t      P.Value    adj.P.Val         B
## 910_at    3.855061  9.660238  29.20918 8.266125e-10 1.043598e-05 11.606193
## 31798_at  3.597334 12.115778  21.04947 1.284430e-08 7.631722e-05  9.890557
## 1854_at   3.340896  8.532099  20.19564 1.813478e-08 7.631722e-05  9.641399
## 38116_at  3.758891  9.513109  16.85669 8.116230e-08 2.511100e-04  8.480197
## 38065_at  2.993641  9.097183  16.20914 1.121213e-07 2.511100e-04  8.214175
## 39755_at  1.765249 12.131839  15.83434 1.359405e-07 2.511100e-04  8.053134
## 1592_at   2.296484  8.311330  15.78841 1.392293e-07 2.511100e-04  8.033025
## 41400_at  2.243510 10.041553  15.28749 1.814762e-07 2.752126e-04  7.808295
## 33730_at -2.041390  8.573470 -15.14298 1.961911e-07 2.752126e-04  7.741556
## 1651_at   2.968283 10.504276  14.78097 2.392480e-07 3.020507e-04  7.570470
```

```
barplot( exprs(eset)["910_at",], las=2, cex.names=.7 )  # top gene
```



```
topTable(fit2,coef=3)
```

```
##                     logFC   AveExpr         t      P.Value    adj.P.Val
## AFFX-CreX-5_at  -6.826151  9.921026 -32.81011 3.108614e-10 2.703546e-06
## AFFX-CreX-3_at  -6.526324 10.406343 -31.58459 4.282844e-10 2.703546e-06
## AFFX-BioDn-5_at -3.734781  8.348424 -19.15515 2.815350e-08 1.184793e-04
## AFFX-BioB-M_at  -3.408360  8.270105 -18.28381 4.143024e-08 1.307642e-04
## AFFX-BioDn-3_at -2.485797 11.401080 -15.52522 1.598695e-07 4.036706e-04
## 39581_at        -2.673757  6.623321 -13.92949 3.886242e-07 6.755348e-04
## AFFX-BioC-3_at  -2.986653  8.159920 -13.92125 3.905047e-07 6.755348e-04
## 37014_at        -1.516742  7.820017 -13.76536 4.280617e-07 6.755348e-04
## 2004_at         -2.064096  6.826916 -12.39873 1.000140e-06 1.402974e-03
## AFFX-BioC-5_at  -2.053876  8.739444 -11.72355 1.570292e-06 1.982494e-03
##                         B
## AFFX-CreX-5_at  10.655112
## AFFX-CreX-3_at  10.547986
## AFFX-BioDn-5_at  8.575319
## AFFX-BioB-M_at   8.336458
## AFFX-BioDn-3_at  7.427475
## 39581_at         6.769730
## AFFX-BioC-3_at   6.766033
## 37014_at         6.695459
## 2004_at          6.021912
## AFFX-BioC-5_at   5.648982
```

# Manual Calculation of logFC and AveExpr

As we already know, the logFC measures the difference in the mean values of gene expression between samples of different conditions without taking into account gene variance. Therefore, given that in our model we have three contrast variables, logFC measures the difference in the mean values of the samples that belong to the conditions included in each contrast variable. In particular for E10, logFC refers to the difference in gene expression between "high10-1.cel" and "low10-1.cel" as well as "high10-2.cel" and "low10-2.cel". For E48 it refers to the difference in gene expression between "high48-1.cel" and "low48-1.cel" as well as "high48-2.cel" and "low48-2.cel". For Time, it refers to "low48-1.cel" and "low10-1.cel" as well as "low48-2.cel" and "low10-2.cel".

In this experiment it can be observed that for each of the 4 different experimental conditions (absent10, present10, absent48, present48) 2 samples were measured. Since 4 samples are numbered with "1" and 4 with "2" we can therefore assume that all samples numbered with "1" belong to a sort of first experiment-trial and all samples numbered with "2" belong to a second one. Thus, in order to calculate the logFC, we can calculate first the differences of gene expression between the samples of each individual experiment and then calculate the mean of the resulting two values. For example, in order to calculate the logFC of the top gene for E10 first we would calculate the differences of gene expression: "high10-1.cel" - "low10-1.cel" as well as "high10-2.cel" - "low10-2.cel". Then we would take the mean of the two resulting values.

Below we calculate the logFC of the top gene for each of the three contrast variables by using the values of our expression set.

```
topgene_coef1 = row.names(topTable(fit2,coef=1)[1, ]);
topgene_coef2 = row.names(topTable(fit2,coef=2)[1, ]);
topgene_coef3 = row.names(topTable(fit2,coef=3)[1, ]);

rowtopgene_coef1 = exprs(eset)[topgene_coef1, ];
rowtopgene_coef2 = exprs(eset)[topgene_coef2, ];
rowtopgene_coef3 = exprs(eset)[topgene_coef3, ];

comp_1 = c(rowtopgene_coef1[4]-rowtopgene_coef1[2],rowtopgene_coef1[3]-rowtopgene_coef1[1]);
LogFC_1 = mean(comp_1); # E10
LogFC_1
```

```
## [1] 2.939428
```

```
comp_2 = c(rowtopgene_coef2[8]-rowtopgene_coef2[6],rowtopgene_coef2[7]-rowtopgene_coef2[5]);
LogFC_2 = mean(comp_2); # E48
LogFC_2
```

```
## [1] 3.855061
```

```
comp_3 = c(rowtopgene_coef3[6]-rowtopgene_coef3[2],rowtopgene_coef3[5]-rowtopgene_coef3[1]);
LogFC_3 = mean(comp_3); # Time
LogFC_3
```

```
## [1] -6.826151
```

Another way that we can use to manually calculate the logFC is to find the mean value of gene expression of the samples that belong to the same condition and then subtract the resulting two mean values between them according to the formula of each contrast variable. This way does not distinguish the samples in different experiments but assumes that the two samples "1", "2" for each condition belong to the same experimental trial. For example, in order to calculate the logFC of the top gene for E10 first we would calculate the mean value of gene expression for each condition: mean value of "high10-1.cel", "high10-2.cel" as well as mean value of "low10-1.cel", "low10-2.cel". Then we would take the difference of the two resulting mean values.

We calculate the logFC of the top gene for each of the three contrast variables and we observe the same results as expected.

```
mean_1 = mean(c(rowtopgene_coef1[3],rowtopgene_coef1[4]))   # E10
mean_2 = mean(c(rowtopgene_coef1[1],rowtopgene_coef1[2]))
LogFC_1 = mean_1 - mean_2
LogFC_1
```

```
## [1] 2.939428
```

```
mean_1 = mean(c(rowtopgene_coef2[7],rowtopgene_coef2[8]))   # E48
mean_2 = mean(c(rowtopgene_coef2[5],rowtopgene_coef2[6]))
LogFC_2 = mean_1 - mean_2
LogFC_2
```

```
## [1] 3.855061
```

```
mean_1 = mean(c(rowtopgene_coef3[5],rowtopgene_coef3[6]))   # Time
mean_2 = mean(c(rowtopgene_coef3[1],rowtopgene_coef3[2]))
LogFC_3 = mean_1 - mean_2
LogFC_3
```

```
## [1] -6.826151
```

In order to calculate manually the average expression of the top gene for each contrast variable we calculate the mean value of all gene expressions across the 8 samples for this specific gene. Instead of using the built in R function mean(), we could sum all indivudual gene expressions and divide this sum by the number of samples.

```
AvExpr_coef1 = mean(rowtopgene_coef1);
AvExpr_coef1
```

```
## [1] 7.876515
```

```
AvExpr_coef2 = mean(rowtopgene_coef2);
AvExpr_coef2
```

```
## [1] 9.660238
```

```
AvExpr_coef3 = mean(rowtopgene_coef3);
AvExpr_coef3
```

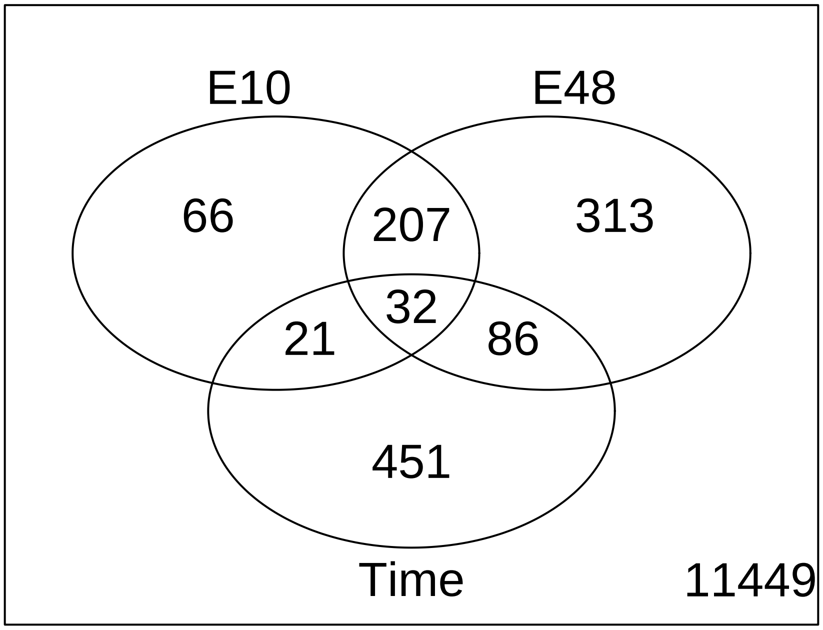
```
## [1] 9.921026
```

We observe that the manually calculated values of logFC and AvExpr out of our expression set are the same with the ones that our model provides.

## Venn Diagram

In order to visualize the numbers of differential genes for ‘E10’, ‘E48’ and ‘Time’ we construct the Venn Diagram. Initially we use the function decideTests which takes as input our model and implements multiple testing procedures for determining whether each statistic in the matrix of t-statistics should be considered significantly different from zero. Then this output is given as input to the function vennDiagram which computes the classification counts and draws the Venn diagram.

```r
res = decideTests(fit2)
vennDiagram(res)
```



Out of the Venn Diagram we summarize the number of genes that are differentiated relatively to each contrast. In particular, we observe that only 66 genes are differentially expressed only in E10, meaning that these genes are differentially expressed in the presence of estrogen for time scale of 10 hours without being differentially expressed in the ansence of estrogen in the same time scale. Moreover, we can see that 207 genes are differentially expressed only when the estrogen is present regardless of the time scale, while 451 genes are differentially expressed only when estrogen is absent in the time scale of 48h but are not differentially expressed in the absence of estrogen for the time scale of 10h. What is worth mentioning is that there are 32 genes which are differentially expressed in E10, E48 and Time, meaning that they are differentially expressed in the presence of estrogen for both time scales as well as in the absence of estrogen for the time scale of 48h. Also we need to mention that the majority of genes (11449) are not classified to one of the three contrasts shown, meaning that they are differentially expressed in conditions which are not captured by the contrast variables that we have defined.

# Alternative Design Matrix

We will try to reproduce the above limma modeling using an alternative design matrix. For this purpose we use the experimental factors defined in the ‘targets’ data frame and construct a design matrix with an intercept column. The intercept column is the absent10 condition which in this alternative way of modeling is the reference condition.

However, as already stated, the form of the design matrix is in accordance with the parameters that need to be estimated. Therefore, by using this alternative design matrix we redefine the parameters of our model which are interpreted differently. Specifically, we define the parameter vector b=transpose([b1 b2 b3 b4]), where b1 denotes the reference condition “absent10”, b2 denotes the condition “absent48” - “absent10”, b3 denotes the condition “present10”-“absent10” and b4 denotes the condition “present48”-“absent10”.

As a result we observe that our parameters are essentially comparisons between our initial conditions. Moreover we can see that the parameters b2 (“absent48” - “absent10”) and b3 (“present10”-“absent10”) form correspondingly the comparisons Time and E10.

In this step we decide not to form a contrast matrix since our current parameters denote contrasts between conditions. By skipping the part of the contrast matrix we fit our model using the limma procedure. Now, since the number of our parameters is four, the numeric matrix containing the estimated coefficients will have four columns. Coef = 1 refers to the reference parameter, coef = 2 refers to the parameter b2 (“absent48” - “absent10”), coef = 3 refers to the parameter b3 (“present10”-“absent10”) and coef = 4 refers to the parameter b4 (“present48”-“absent10”). Since parameters b2 and b3 form the comparisons modeled by the contrast variables Time and E10 of our previous model correspondingly we expect to have the same computed statistics and the same top genes. We see that this is the case.

```r
design_mine <- model.matrix(~f)
colnames(design_mine) <- levels(f)
design_mine
```

```
##   absent10 absent48 present10 present48
## 1        1        0         0         0
## 2        1        0         0         0
## 3        1        0         1         0
## 4        1        0         1         0
## 5        1        1         0         0
## 6        1        1         0         0
## 7        1        0         0         1
## 8        1        0         0         1
## attr(,"assign")
## [1] 0 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$f
## [1] "contr.treatment"
```

```r
fit_mine <- lmFit(eset, design_mine)
fit2_mine <- eBayes(fit_mine)

topTable(fit2_mine,coef=2)
```

```
##                     logFC   AveExpr         t      P.Value    adj.P.Val
## AFFX-CreX-5_at  -6.826151  9.921026 -32.81011 3.108614e-10 2.703546e-06
## AFFX-CreX-3_at  -6.526324 10.406343 -31.58459 4.282844e-10 2.703546e-06
## AFFX-BioDn-5_at -3.734781  8.348424 -19.15515 2.815350e-08 1.184793e-04
## AFFX-BioB-M_at  -3.408360  8.270105 -18.28381 4.143024e-08 1.307642e-04
## AFFX-BioDn-3_at -2.485797 11.401080 -15.52522 1.598695e-07 4.036706e-04
## 39581_at        -2.673757  6.623321 -13.92949 3.886242e-07 6.755348e-04
## AFFX-BioC-3_at  -2.986653  8.159920 -13.92125 3.905047e-07 6.755348e-04
## 37014_at        -1.516742  7.820017 -13.76536 4.280617e-07 6.755348e-04
## 2004_at         -2.064096  6.826916 -12.39873 1.000140e-06 1.402974e-03
## AFFX-BioC-5_at  -2.053876  8.739444 -11.72355 1.570292e-06 1.982494e-03
##                         B
## AFFX-CreX-5_at  10.655112
## AFFX-CreX-3_at  10.547986
## AFFX-BioDn-5_at  8.575319
## AFFX-BioB-M_at   8.336458
## AFFX-BioDn-3_at  7.427475
## 39581_at         6.769730
## AFFX-BioC-3_at   6.766033
## 37014_at         6.695459
## 2004_at          6.021912
## AFFX-BioC-5_at   5.648982
```

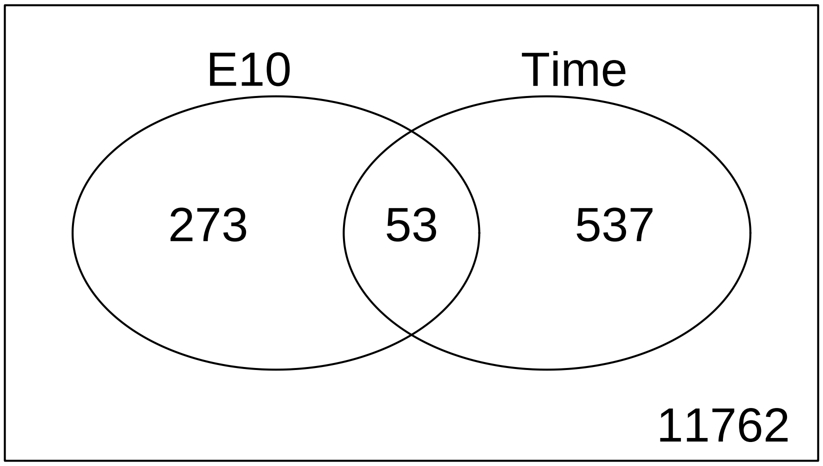
```
topTable(fit2_mine,coef=3)
```

```
##              logFC   AveExpr        t      P.Value    adj.P.Val        B
## 39642_at  2.939428  7.876515 23.71715 4.741579e-09 3.128295e-05 9.966810
## 910_at    3.113733  9.660238 23.59225 4.955715e-09 3.128295e-05 9.942522
## 31798_at  2.800195 12.115778 16.38509 1.025747e-07 3.511070e-04 7.977290
## 41400_at  2.381040 10.041553 16.22463 1.112418e-07 3.511070e-04 7.916921
## 40117_at  2.555282  9.676557 15.68070 1.472942e-07 3.576234e-04 7.705093
## 1854_at   2.507616  8.532099 15.15848 1.945518e-07 3.576234e-04 7.490766
## 39755_at  1.679331 12.131839 15.06365 2.048314e-07 3.576234e-04 7.450643
## 1824_s_at 1.914637  9.238870 14.87915 2.266129e-07 3.576234e-04 7.371475
## 1126_s_at 1.782825  6.879918 13.83040 4.119252e-07 5.778395e-04 6.892307
## 1536_at   2.662258  5.937222 13.26247 5.795111e-07 7.316327e-04 6.610486
```

In this model the parameter b4 forms a comparison ("present48"-"absent10") which we did not have in the previous model. Respectively, the contrast variable E48 of our previous model is not formed in the alternative modeling that we have used. Therefore, we see that our new model reproduces our initial one in 2 coefficients and not in 3.

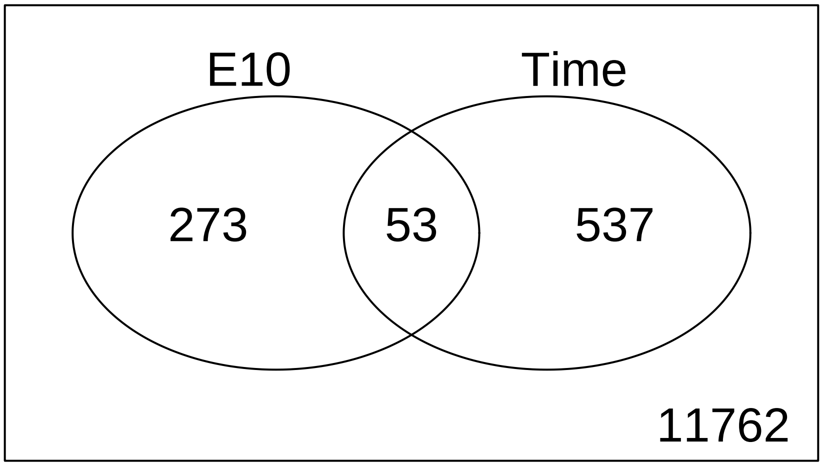
In the Venn Diagram shown below we vizualize the number of differential genes for the parameters b2 ("absent48" - "absent10") and b3 ("present10"-"absent10"). Taking into account the correspondence of these two parameters with Time and E10 respectively we construct the following Venn Diagram.

```
res_mine = decideTests(fit2_mine)
group_coef = cbind(res_mine[ ,"present10"],res_mine[,"absent48"])
colnames(group_coef) = c("E10","Time");
vennDiagram(group_coef)
```



Now, we plot the Venn Digram of our initial model only for Time and E10 and we receive the same Diagram as expected.

```r
group_coef_initial = cbind(res[ ,"E10"],res[,"Time"])
colnames(group_coef_initial) = c("E10","Time");
vennDiagram(group_coef_initial)
```



As already stated, this new model reproduces our initial one in 2 coefficients (E10, Time) and not in 3. In order to fully reproduce our initial model, we need to model also the contrast E48. Since this contrast cannot be modeled directly from our design matrix (design_mine) we will add a contrast matrix. This contrast matrix takes into account that our parameters already form comparisons, therefore it will include the parameters b2, b3 and the contrast b4-b2 ("present48" - "absent10" - ("absent48" - "absent10") = "present48" - "absent48") which results in E48.

By keeping the design matrix as it is, the contrast matrix is constructed as follows. Then we "enter" our contrast matrix in the model.

```
cont.matrix_mine <- makeContrasts(E10="present10",E48="present48-absent48",
                                  TIME ="absent48",levels=design_mine)

cont.matrix_mine
```

```
##              Contrasts
## Levels        E10 E48 TIME
##    absent10     0   0    0
##    absent48     0  -1    1
##    present10    1   0    0
##    present48    0   1    0
```

```
fit2_betw  <- contrasts.fit(fit_mine, cont.matrix_mine)
fit2_mine2<- eBayes(fit2_betw)

topTable(fit2_mine2,coef=1)
```

```
##               logFC   AveExpr        t      P.Value    adj.P.Val        B
## 39642_at  2.939428  7.876515 23.71715 4.741579e-09 3.128295e-05 9.966810
## 910_at    3.113733  9.660238 23.59225 4.955715e-09 3.128295e-05 9.942522
## 31798_at  2.800195 12.115778 16.38509 1.025747e-07 3.511070e-04 7.977290
## 41400_at  2.381040 10.041553 16.22463 1.112418e-07 3.511070e-04 7.916921
## 40117_at  2.555282  9.676557 15.68070 1.472942e-07 3.576234e-04 7.705093
## 1854_at   2.507616  8.532099 15.15848 1.945518e-07 3.576234e-04 7.490766
## 39755_at  1.679331 12.131839 15.06365 2.048314e-07 3.576234e-04 7.450643
## 1824_s_at 1.914637  9.238870 14.87915 2.266129e-07 3.576234e-04 7.371475
## 1126_s_at 1.782825  6.879918 13.83040 4.119252e-07 5.778395e-04 6.892307
## 1536_at   2.662258  5.937222 13.26247 5.795111e-07 7.316327e-04 6.610486
```

```
topTable(fit2_mine2,coef=2)
```

```
##               logFC   AveExpr         t      P.Value    adj.P.Val         B
## 910_at    3.855061  9.660238  29.20918 8.266125e-10 1.043598e-05 11.606193
## 31798_at  3.597334 12.115778  21.04947 1.284430e-08 7.631722e-05  9.890557
## 1854_at   3.340896  8.532099  20.19564 1.813478e-08 7.631722e-05  9.641399
## 38116_at  3.758891  9.513109  16.85669 8.116230e-08 2.511100e-04  8.480197
## 38065_at  2.993641  9.097183  16.20914 1.121213e-07 2.511100e-04  8.214175
## 39755_at  1.765249 12.131839  15.83434 1.359405e-07 2.511100e-04  8.053134
## 1592_at   2.296484  8.311330  15.78841 1.392293e-07 2.511100e-04  8.033025
## 41400_at  2.243510 10.041553  15.28749 1.814762e-07 2.752126e-04  7.808295
## 33730_at -2.041390  8.573470 -15.14298 1.961911e-07 2.752126e-04  7.741556
## 1651_at   2.968283 10.504276  14.78097 2.392480e-07 3.020507e-04  7.570470
```

```
topTable(fit2_mine2,coef=3)
```

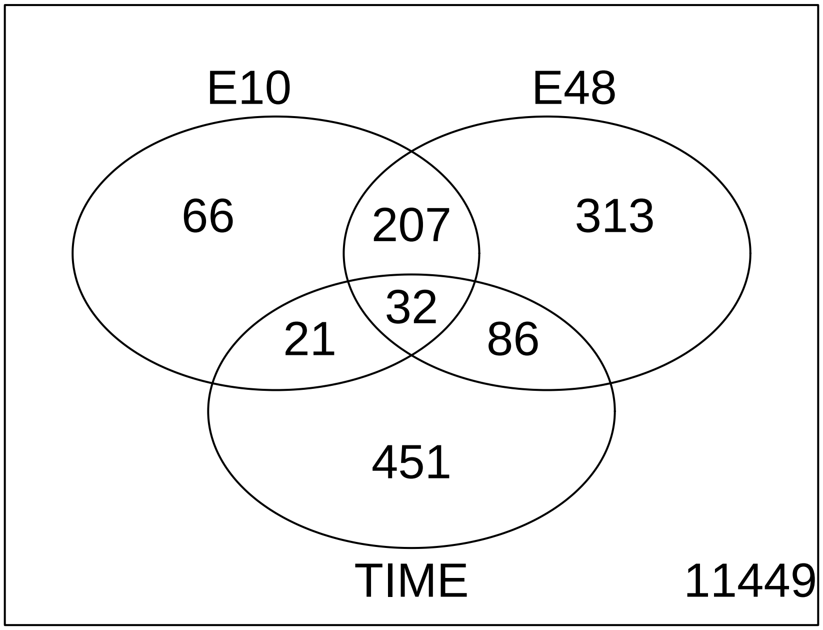
```
##                      logFC   AveExpr         t      P.Value    adj.P.Val
## AFFX-CreX-5_at  -6.826151  9.921026 -32.81011 3.108614e-10 2.703546e-06
## AFFX-CreX-3_at  -6.526324 10.406343 -31.58459 4.282844e-10 2.703546e-06
## AFFX-BioDn-5_at -3.734781  8.348424 -19.15515 2.815350e-08 1.184793e-04
## AFFX-BioB-M_at  -3.408360  8.270105 -18.28381 4.143024e-08 1.307642e-04
## AFFX-BioDn-3_at -2.485797 11.401080 -15.52522 1.598695e-07 4.036706e-04
## 39581_at        -2.673757  6.623321 -13.92949 3.886242e-07 6.755348e-04
## AFFX-BioC-3_at  -2.986653  8.159920 -13.92125 3.905047e-07 6.755348e-04
## 37014_at        -1.516742  7.820017 -13.76536 4.280617e-07 6.755348e-04
## 2004_at         -2.064096  6.826916 -12.39873 1.000140e-06 1.402974e-03
## AFFX-BioC-5_at  -2.053876  8.739444 -11.72355 1.570292e-06 1.982494e-03
##                          B
## AFFX-CreX-5_at  10.655112
## AFFX-CreX-3_at  10.547986
## AFFX-BioDn-5_at  8.575319
## AFFX-BioB-M_at   8.336458
## AFFX-BioDn-3_at  7.427475
## 39581_at         6.769730
## AFFX-BioC-3_at   6.766033
## 37014_at         6.695459
## 2004_at          6.021912
## AFFX-BioC-5_at   5.648982
```

Now the number of columns in the coefficients matrix is 3. Coef = 1 refers to b3 (E10, “present10” - “absent10”), coef = 2 refers to the comparison b4-b2 (E48, “present48”-“absent48”) and coef = 3 refers to the parameter b2 (Time, “absent48”-“absent10”).

Since our newest model reproduces all three contrasts of our initial model, we receive the same statistics with our initial model for all three contrasts. As expected the top genes of our model for each contrast are reproduced.

Additionally we plot the Venn Diagram and we observe the same classification of genes as in our initial model.

```
res_mine2 = decideTests(fit2_mine2)
vennDiagram(res_mine2)
```



To conclude, our newest alternative model reproduces fully the initial one.