

# thema07

keimpe dijkstra

8-3-2022

## 1 Exploratory data analysis

```
imports
library(dplyr)

##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##     filter, lag
## The following objects are masked from 'package:base':
##     intersect, setdiff, setequal, union
library(affy)

## Loading required package: BiocGenerics
## Loading required package: parallel
##
## Attaching package: 'BiocGenerics'
## The following objects are masked from 'package:parallel':
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
## The following objects are masked from 'package:dplyr':
##     combine, intersect, setdiff, union
## The following objects are masked from 'package:stats':
##     IQR, mad, sd, var, xtabs
## The following objects are masked from 'package:base':
##     Filter, Find, Map, Position, Reduce, anyDuplicated, append,
##     as.data.frame, basename, cbind, colnames, dirname, do.call,
##     duplicated, eval, evalq, get, grep, grepl, intersect, is.unsorted,
##     lapply, mapply, match, mget, order, paste, pmax, pmax.int, pmin,
##     pmin.int, rank, rbind, rownames, sapply, setdiff, sort, table,
##     tapply, union, unique, unsplit, which, which.max, which.min
```

```

## Loading required package: Biobase
## Welcome to Bioconductor
##
##      Vignettes contain introductory material; view with
##      'browseVignettes()'. To cite Bioconductor, see
##      'citation("Biobase")', and for packages 'citation("pkgname")'.
library(scales)
library(DESeq2)

## Loading required package: S4Vectors
## Loading required package: stats4
##
## Attaching package: 'S4Vectors'
## The following objects are masked from 'package:dplyr':
##   first, rename
## The following object is masked from 'package:base':
##   expand.grid
## Loading required package: IRanges
##
## Attaching package: 'IRanges'
## The following objects are masked from 'package:dplyr':
##   collapse, desc, slice
## Loading required package: GenomicRanges
## Loading required package: GenomeInfoDb
## Loading required package: SummarizedExperiment
## Loading required package: DelayedArray
## Loading required package: matrixStats
##
## Attaching package: 'matrixStats'
## The following objects are masked from 'package:Biobase':
##   anyMissing, rowMedians
## The following object is masked from 'package:dplyr':
##   count
##
## Attaching package: 'DelayedArray'
## The following objects are masked from 'package:matrixStats':
##   colMaxs, colMins, colRanges, rowMaxs, rowMins, rowRanges

```

```

## The following objects are masked from 'package:base':
##
##     aperm, apply, rowsum

library(pheatmap)
library(PoiClaClu)
library(ggplot2)
library(edgeR)

## Loading required package: limma

##
## Attaching package: 'limma'

## The following object is masked from 'package:DESeq2':
##
##     plotMA

## The following object is masked from 'package:BiocGenerics':
##
##     plotMA

library(EnhancedVolcano)

## Loading required package: ggrepel

library(gplots)

##
## Attaching package: 'gplots'

## The following object is masked from 'package:IRanges':
##
##     space

## The following object is masked from 'package:S4Vectors':
##
##     space

## The following object is masked from 'package:stats':
##
##     lowess

```

## Contents

<b>1 Exploratory data analysis</b>	<b>1</b>
1.1 visualizations . . . . .	4
1.1.1 heatmap j147 . . . . .	9
1.1.2 heatmap cad-31 . . . . .	10
1.1.3 multi dimensional scaling plot . . . . .	11
1.1.4 conclusions EDA . . . . .	12
1.2 Fold Change . . . . .	14
1.2.1 j47 old vs old . . . . .	14
1.2.2 AD.cad31 vs AD . . . . .	15
1.3 Bio conductor . . . . .	16

First, we load all our data into r

```

#load in the data using the read.table method
j147 <- read.table("./data/j147.csv", header = T, sep = ",", quote="")
cad31 <- read.table("./data/cad-31.txt", header = T, sep = "\t", fill = T, quote="")

#create a new column for te gene annotation
cad31$Gene <- strsplit(cad31$Annotation.Divergence, "|", 1)

#filter the gene annotation out of the string
counter <- 1
while( counter < length(cad31$Gene)) {

  cad31$Gene[counter] <- cad31$Gene[[counter]][1]
  counter <- counter + 1
}

#drop some unnecesary columns
cad31 <- cad31[c(22, 9:21)]

#merge all the data into one dataframe
data <- merge(j147, cad31)

#replace na's with zero's
data[is.na(data)] <- 0

#rename the columns
names(data) <- c(
  paste0('Gene'),
  paste0('AD.old.j147.', 1:3),
  paste0('AD.old.', 1:3),
  paste0('AD.young.', 1:4),
  paste0('AD.cad31.', 1:3),
  paste0('AD.', 1:3),
  paste0('WT.CAD31.', 1:3),
  paste0('WT.', 1:4)
)

#TODO: Check row names just to be sure

#set some indices to help future work
AD.old.j147 <- 2:4
AD.old <- 5:7
AD.young <- 8:11
AD.cad31 <- 12:14
AD <- 15:17
WT.CAD31 <- 18:20
WT <- 21:24

```

## 1.1 visualizations

basic statistics

```
summary(rowSums(data[AD.old.j147])/3)
```

```
##      Min.    1st Qu.     Median      Mean    3rd Qu.      Max.
```

```

##      0.00     0.33    56.67   580.69   498.33 87522.00
summary(rowSums(data[AD.old])/3)

##      Min. 1st Qu. Median     Mean 3rd Qu.     Max.
##      0.00    0.33   53.00   540.94  464.67 76552.67

summary(rowSums(data[AD.young])/4)

##      Min. 1st Qu. Median     Mean 3rd Qu.     Max.
##      0.00    0.00   41.25   446.74  382.00 76737.25

summary(rowSums(data[AD.cad31])/3)

##      Min. 1st Qu. Median     Mean 3rd Qu.     Max.
##      0.00    0.33   62.67   585.45  472.00 69317.33

summary(rowSums(data[AD])/3)

##      Min. 1st Qu. Median     Mean 3rd Qu.     Max.
##      0.00    0.33   55.67   499.38  410.00 63741.33

summary(rowSums(data[WT.CAD31])/3)

##      Min. 1st Qu. Median     Mean 3rd Qu.     Max.
##      0.00    0.33   44.00   412.07  342.00 43685.33

summary(rowSums(data[WT])/4)

##      Min. 1st Qu. Median     Mean 3rd Qu.     Max.
##      0.0     0.5    63.0    633.5   508.2 71066.2

```

The min and first quantile all show similar or very similar results. The wildtype mouse shows the most expression with an almost double maximum than its drugged counterpart.

The young mouse and the mouse on CAD-31 have the least sequences read whereas the mouse with AD and on drugs seem to be upregulated.

This might be explained because of regulation but also the testing can have an influence on the amount of sequences that are read, therefore we need to normalize the data.

Add 1 to the whole dataframe so it can logscale

```

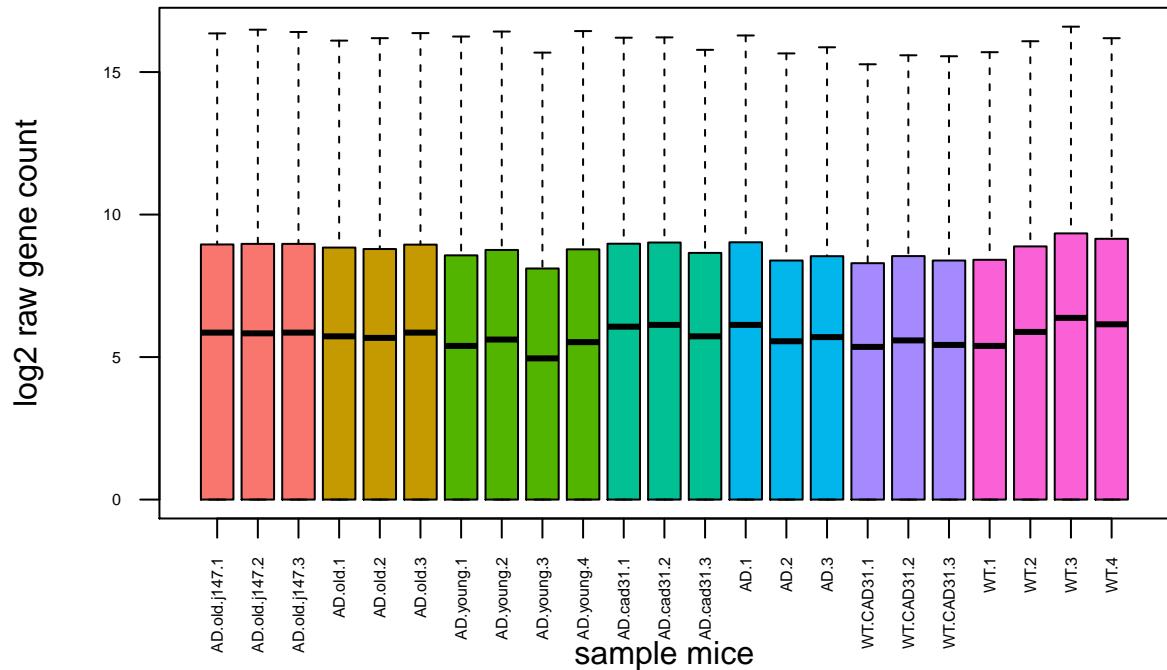
#set the gene column as row names
row.names(data) <- data$Gene
data[1] <- NULL

data <- data + 1

Boxplot
myColors <- hue_pal()(7)

boxplot(log2(data), las=2, xlab = "sample mice",
       ylab = "log2 raw gene count", cex.axis=0.5,
       col=rep(myColors,c(3,3,4,3,3,3,4)))

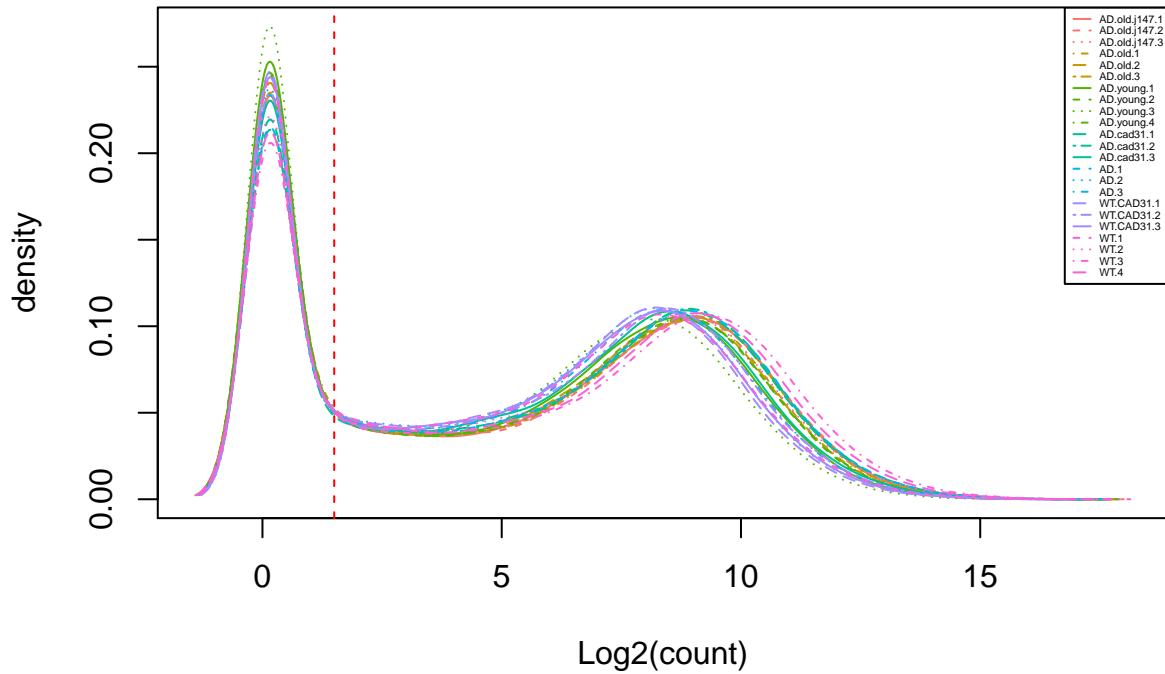
```



Density plot

```
## Plot the log2-transformed data with a 0.1 pseudocount
plotDensity(log2(data + 0.1),
            col=rep(myColors,c(3,3,4,3,3,3,4)),
            lty=c(1:ncol(data)), xlab='Log2(count)',
            main='Expression Distribution')
legend('topright', names(data), lty=c(1:ncol(data)),
       col=rep(myColors,c(3,3,4,3,3,3,4)), cex = 0.3)
abline(v=1.5, lwd=1, col='red', lty=2)
```

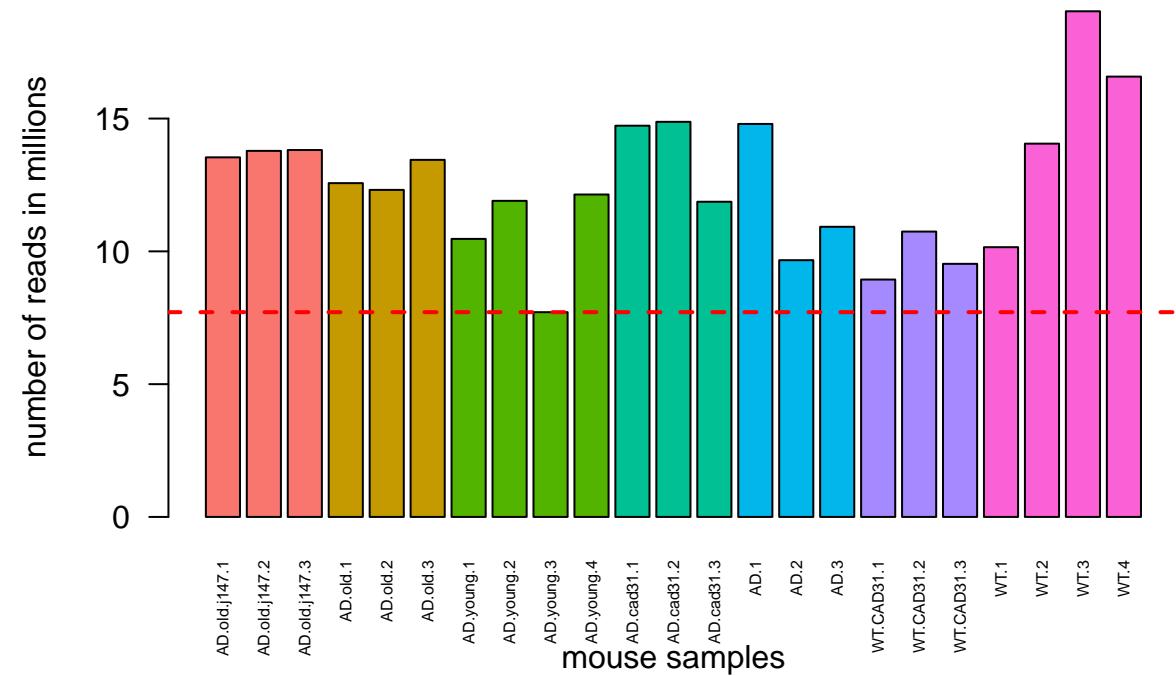
## Expression Distribution



Barplot sequence depth

```
barplot(colSums(data)/1000000,      col=rep(myColors,c(3,3,4,3,3,3,4)),
       xlab = "mouse samples", ylab = "number of reads in millions", cex.names= 0.5, las=2)

abline(h = min(colSums((data)/1000000)), col = "red",
       lty=2, lwd=2)
```



normalization

```

(ddsMat <- DESeqDataSetFromMatrix(countData = data,
                                    colData = data.frame(samples = names(data)),
                                    design = ~ 1))

## converting counts to integer mode

## class: DESeqDataSet
## dim: 23573 23
## metadata(1): version
## assays(1): counts
## rownames(23573): 0610005C13Rik 0610007P14Rik ... a 17Rn6
## rowData names(0):
## colnames(23): AD.old.j147.1 AD.old.j147.2 ... WT.3 WT.4
## colData names(1): samples

rld.dds <- vst(ddsMat)

rld <- assay(rld.dds)

j.av <- mean(rld[,1:10])
c.av <- mean(rld[,11:23])

j.bi <- mean(rld[,1:10]) +qt(c(0.025, 0.975), length(rld[,1:10])-1)*sd(rld[,1:10])/sqrt(length(rld[,1:10]))

c.bi <- mean(rld[,11:23]) +qt(c(0.025, 0.975), length(rld[,11:23])-1)*sd(rld[,11:23])/sqrt(length(rld[,11:23]))

#rld[,11:23] <-rld[,11:23] * (c.av/j.av)

sampledist <- dist(t(rld), method = "euclidean", diag = T, upper = T)
sampledistmatrix <- as.matrix(sampledist)

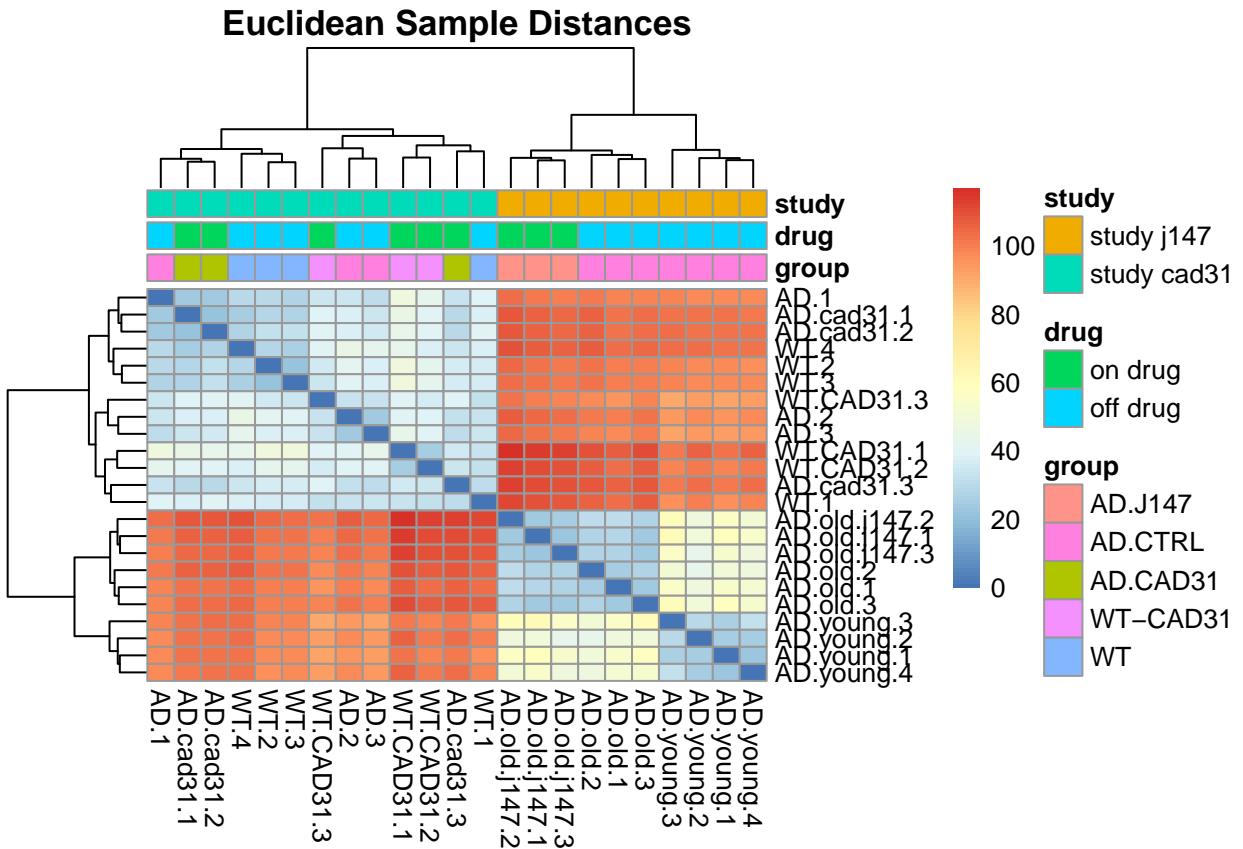
heatmap(both datasets)

ann <- data.frame(group = factor(c(1,1,1,rep(2,each=7),3,3,3,4,4,4,5,5,5,6,6,6,6),
                                   labels = c("AD.J147", "AD.CTRL", "AD.CAD31", "AD.CTRL", "WT-CAD31", "WT"),
                                   drug = factor(c(rep.int(1,3), rep.int(2,7), rep.int(1,3), rep.int(2,3), rep.int(1,3),
                                   study = factor(c(rep.int(1,10), rep.int(2,13)),
                                   labels = c("study j147", "study cad31"))))

row.names(ann) <- names(data)

pheatmap(sampledistmatrix, show_colnames = T,
        annotation_col = ann,
        clustering_distance_rows = sampledist,
        clustering_distance_cols = sampledist,
        main = "Euclidean Sample Distances")

```



### 1.1.1 heatmap j147

```

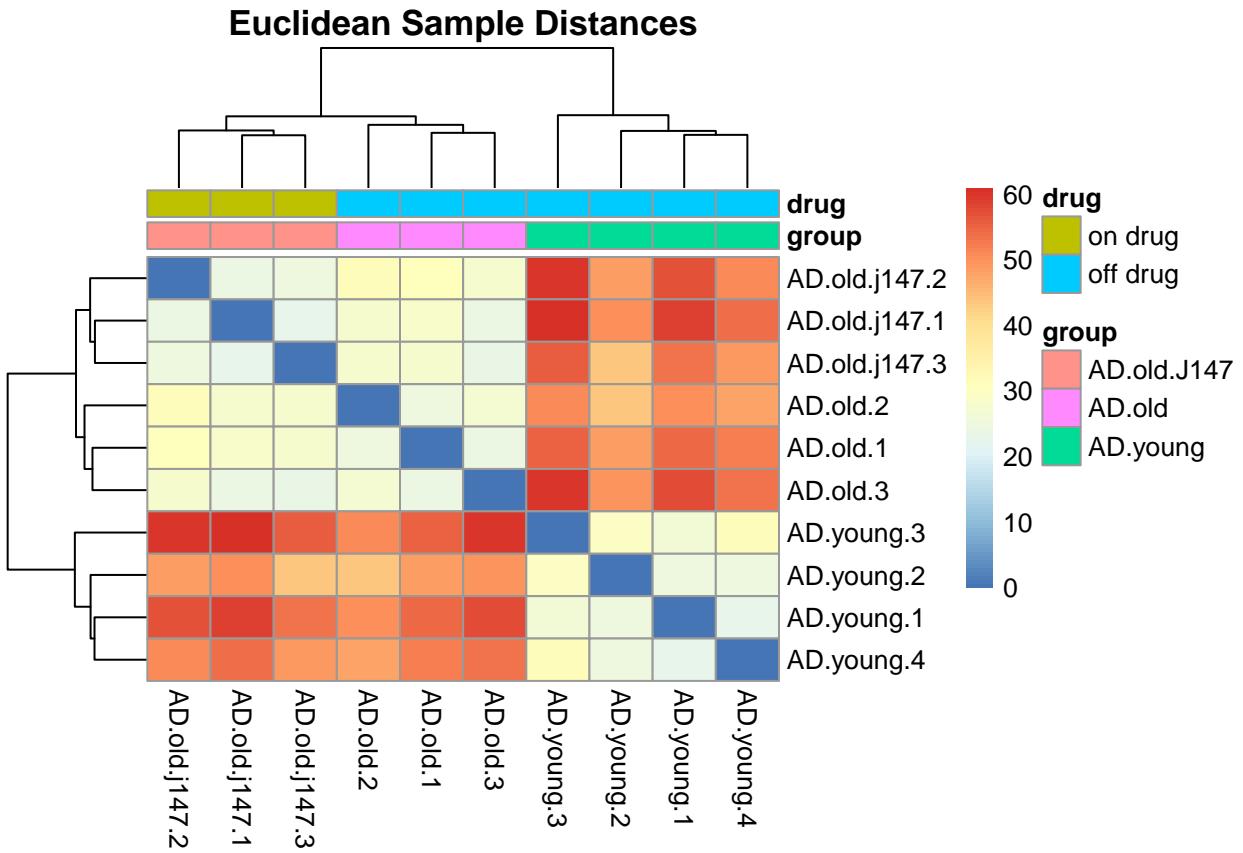
ann <- data.frame(group = factor(c(1,1,1,rep(2,each=3), rep(3, each=4)), labels = c("AD.old.J147", "AD.young.J147", "control.J147", "AD.old.J148", "AD.young.J148", "control.J148", "AD.old.J149", "AD.young.J149", "control.J149", "AD.old.J150", "AD.young.J150", "control.J150)), drug = factor(c(rep.int(1,3), rep.int(2,7)), labels = c("on drug", "off drug")))

row.names(ann) <- names(data[1:10])

sampledist <- dist(t(rld[,1:10]), method = "euclidean", diag = T, upper = T)
sampledistmatrix <- as.matrix(sampledist)

pheatmap(sampledistmatrix, show_colnames = T,
         annotation_col = ann,
         clustering_distance_rows = sampledist,
         clustering_distance_cols = sampledist,
         main = "Euclidean Sample Distances")

```



### 1.1.2 heatmap cad-31

```

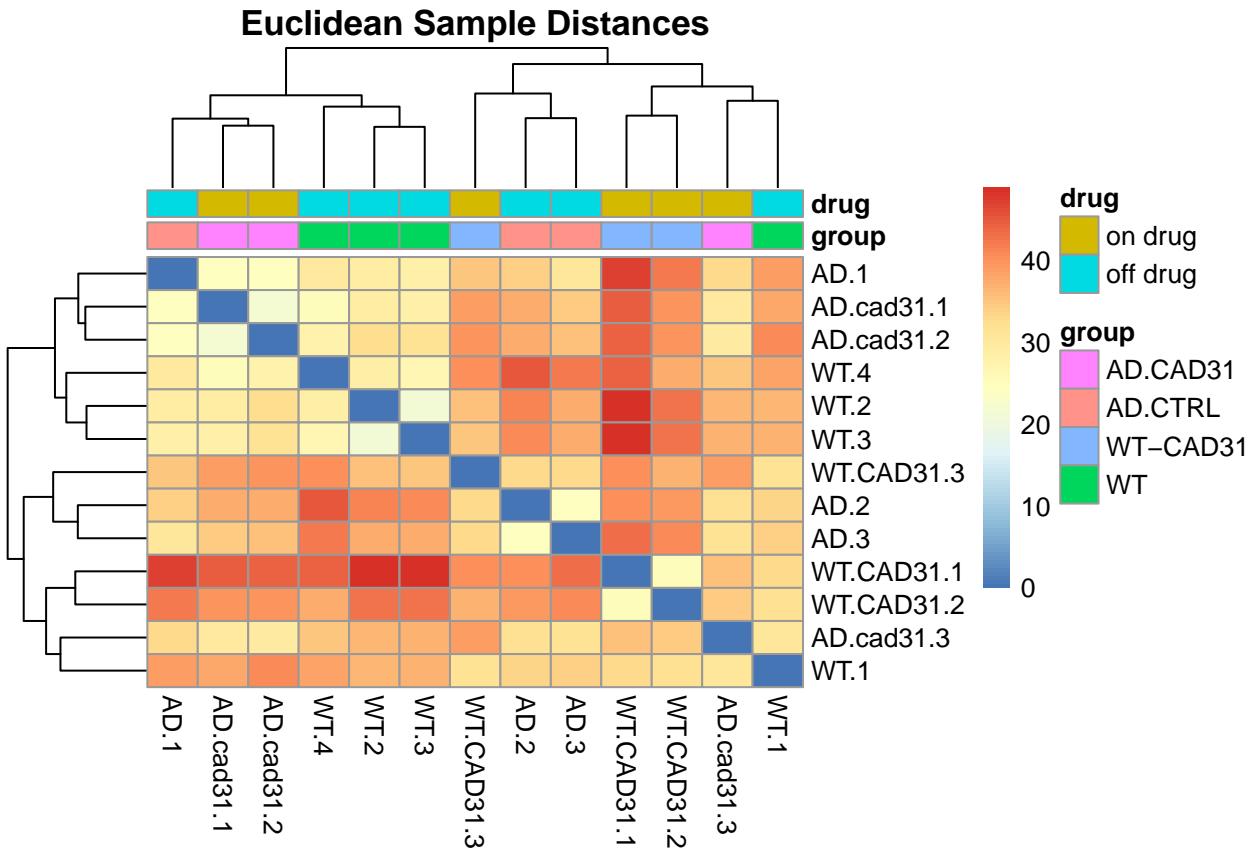
ann <- data.frame(group = factor(c(rep(1,each=3),rep(2,each=3), rep(3, each=3), rep(4,each=4)), labels =
                                drug = factor(c(rep.int(1,3), rep.int(2,3), rep.int(1,3), rep.int(2,4)), labels = c("on drug", "off drug"))

row.names(ann) <- names(data[11:23])

sampledist <- dist(t(rld[,11:23]), method = "euclidean", diag = T, upper = T)
sampledistmatrix <- as.matrix(sampledist)

pheatmap(sampledistmatrix, show_colnames = T,
         annotation_col = ann,
         clustering_distance_rows = sampledist,
         clustering_distance_cols = sampledist,
         main = "Euclidean Sample Distances")

```



### 1.1.3 multi dimensional scaling plot

```

dds <- assay(ddsMat)
poisd <- PoissonDistance( t(dds), type = "deseq")
# Extract the matrix with distances
samplePoisDistMatrix <- as.matrix(poisd$dd)
# Calculate the MDS and get the X- and Y-coordinates
mdsPoisData <- data.frame( cmdscale(samplePoisDistMatrix) )

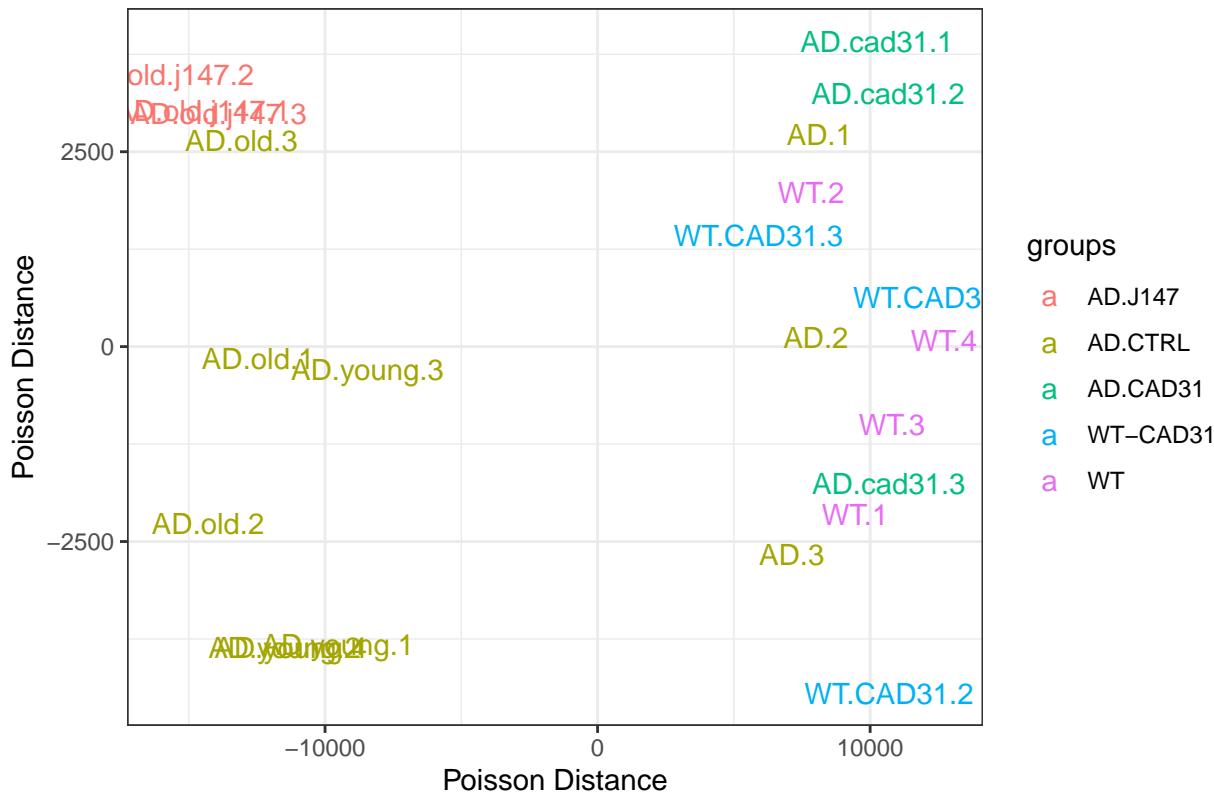
# And set some better readable names for the columns
names(mdsPoisData) <- c('x_coord', 'y_coord')
groups <- factor(c(1,1,1,rep(2,each=7),3,3,3,4,4,4,5,5,5,6,6,6,6),
                  labels = c("AD.J147", "AD.CTRL", "AD.CAD31", "AD.CTRL", "WT-CAD31", "WT"))

coldata <- names(data)

# Create the plot using ggplot
ggplot(mdsPoisData, aes(x_coord, y_coord, color = groups, label = coldata)) +
  geom_text(size = 4) +
  ggtitle('Multi Dimensional Scaling') +
  labs(x = "Poisson Distance", y = "Poisson Distance") +
  theme_bw()

```

## Multi Dimensional Scaling



### 1.1.4 conclusions EDA

The first thing that pops out when looking at the images created is that there are big differences between the two datasets. This can be seen in the heatmap containing both studies and in the mds plot as well. This throws our comparison of gene expression a bit off, but it shouldn't be too big of a problem because we can just compare the DEG's later on.

A second thing that can be noticed in the heatmap of the j147 is that the age of the mouse seems of high influence on the gene expression but in the mds plot there also is a noticeable difference between mice on and off the drug.

The cad-31 study seems to produce some more mixed results, although harder to detect the same pattern as in the j147 seems to apply.

```
# Perform a naive FPM normalization
# Note: log transformation includes a pseudocount of 1
counts.fpm <- log2( (data / (colSums(data) / 1e6)) + 1 )

num.below.one <- 0
num.below.two <- 0

means <- rowMeans(counts.fpm)
difference <- c()

num.with.zero <- 0
num.with.one <- 0
num.with.zero.rows <- c()

i <- 1
```

```

while (i <= length(means)){
  if(means[i] < 1){
    num.below.one<- num.below.one+ 1
  }
  if(means[i] < 2){
    num.below.two <- num.below.two +1
  }
  j <- 1
  pass <- F

  while(j < length(counts.fpm[i,])){
    if(counts.fpm[i,j] < 1 & pass == F){
      num.with.zero <- num.with.zero +1
      pass <- T
      num.with.zero.rows <- append(num.with.zero.rows, i)
    }
    if(counts.fpm[i,j] < 2){
      num.with.one <- num.with.one +1
      j <- length(counts.fpm[i,])
    }
    j <- j+1
  }

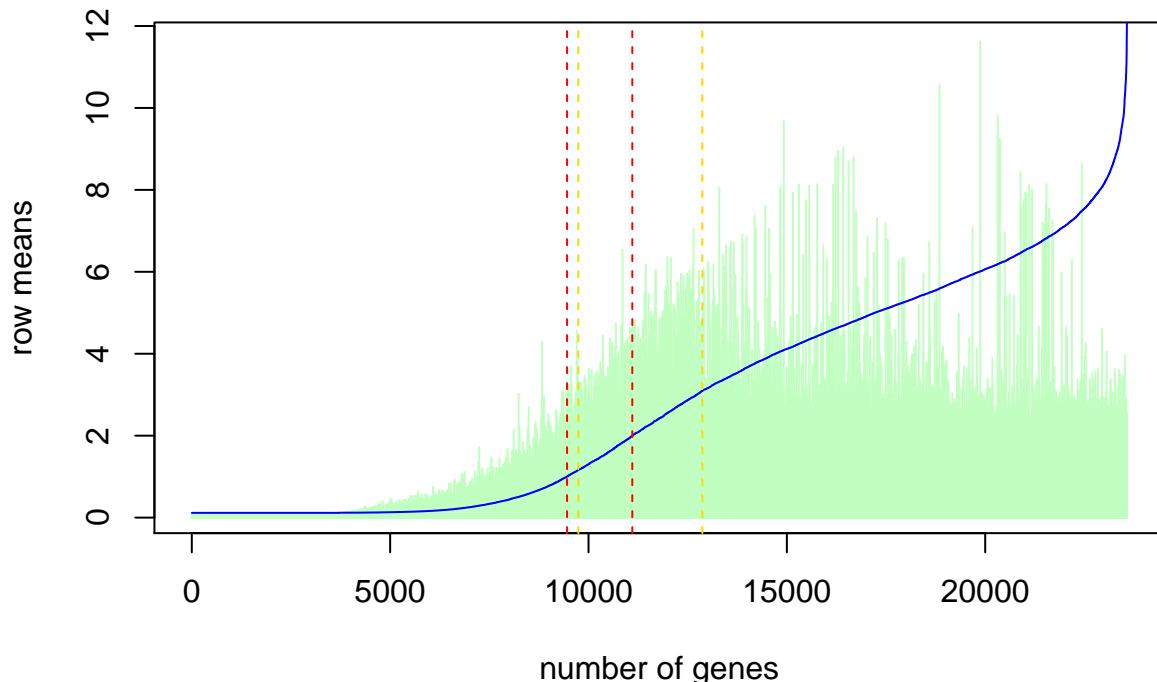
  difference <- append(difference, max(counts.fpm[i,])-min(counts.fpm[i,]))

  i <- i+1}

means <- as.data.frame(means)
means <- cbind(means, difference)
means <- means[order(means[,1]),]

plot(means$difference,type="h",xlab="number of genes", ylab="row means", col="darkseagreen1")
lines(means$means,type="l" ,col="blue",)
abline(v=num.below.one, col="red", lty=2)
abline(v=num.below.two, col="red", lty=2)
abline(v=num.with.zero, col="gold", lty=2)
abline(v=num.with.one, col="gold", lty=2)

```



The plot above describes the means of the rows in the normalized data. The abline's in red are values where the means are below 1 and two respectively and the yellow abline's rows that contain values below 1 and 2. The small difference between the abline's below 1 suggests that there isn't much of a difference between the values, the bigger gap between the values of two suggest the opposite.

The difference is seen in the light green histogram in the back, it first shows no difference relative to the means value, but as the mean gets bigger the difference also starts to increase and eventually kinda settles.

Therefore i decide to set the threshold of filtering where none of the values in the row is below one since after that threshold there seems to be a incline of the row means.

```
original_length <- nrow(counts.fpm)
counts.fpm <- counts.fpm[-num.with.zero.rows,]
cat(original_length-nrow(counts.fpm), "rows are deleted")
## 9743 rows are deleted
```

## 1.2 Fold Change

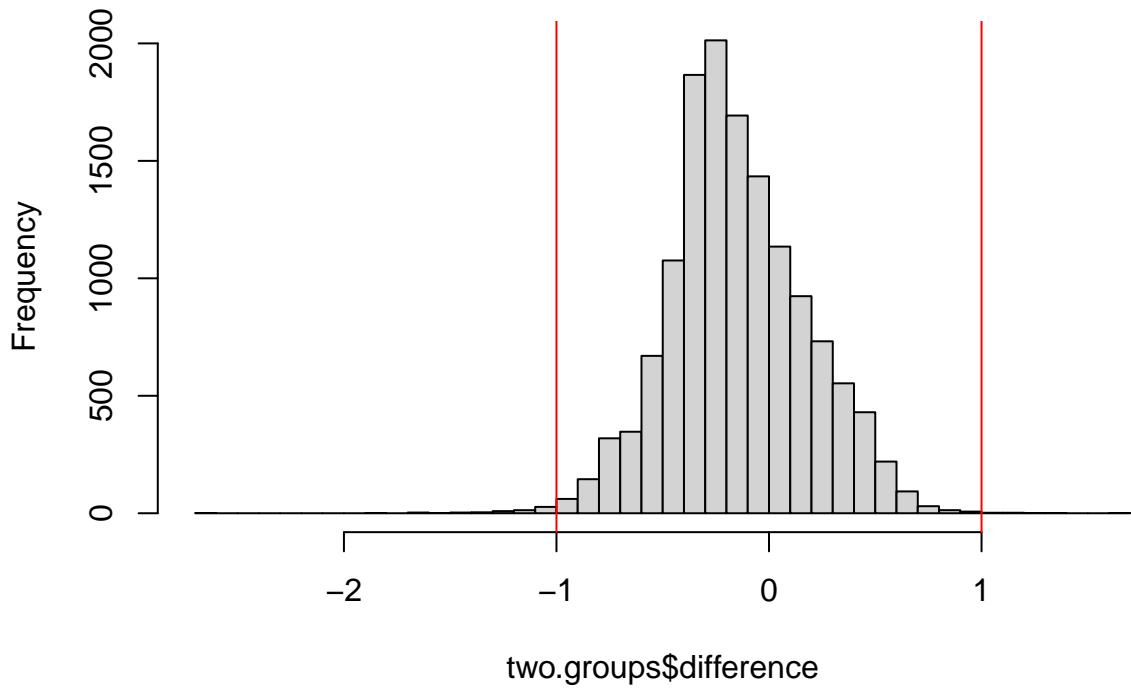
### 1.2.1 j47 old vs old

```
two.groups <- as.data.frame(rowMeans(counts.fpm[AD.old.j147]))
two.groups["AD.old1"] <- rowMeans(counts.fpm[AD.old])
names(two.groups)[1] <- "AD.old.j147"

two.groups["difference"] <- two.groups$AD.old1 - two.groups$AD.old.j147

hist(two.groups$difference, breaks=60, main = "histogram of fold change difference")
abline(v=-1, col="red")
abline(v=1, col="red")
```

## histogram of fold change difference



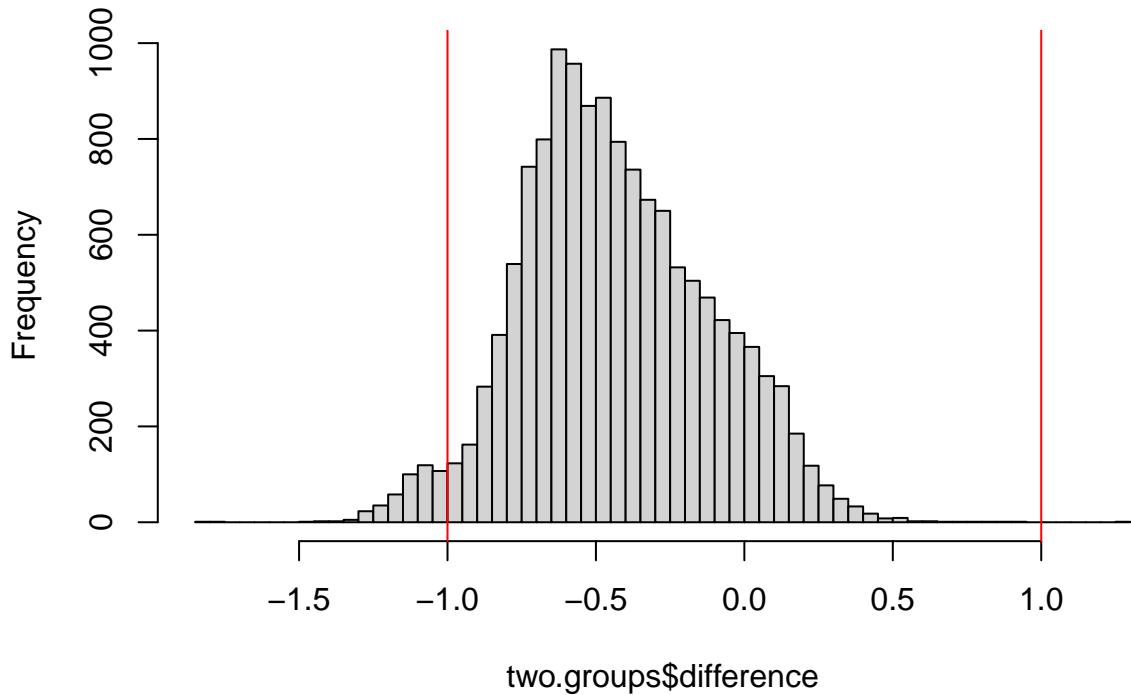
### 1.2.2 AD.cad31 vs AD

```
two.groups <- as.data.frame(rowMeans(counts.fpm[AD.cad31]))
two.groups["AD"] <- rowMeans(counts.fpm[AD])
names(two.groups)[1] <- "AD.cad31"

two.groups["difference"] <- two.groups$AD - two.groups$AD.cad31

hist(two.groups$difference, breaks=60, main = "histogram of fold change difference")
abline(v=-1, col="red")
abline(v=1, col="red")
```

histogram of fold change difference



### 1.3 Bio conductor

```
group <- factor(c(1,1,1,rep(2,each=3),3,3,3,3,4,4,4,5,5,5,6,6,6,rep(7, each=4)),
                  labels = c("AD.J147", "AD.CTRL.OLD","AD.CTRL.YOUNG", "AD.CAD31", "AD.CT
(design <- model.matrix(~ group))

##      (Intercept) groupAD.CTRL.OLD groupAD.CTRL.YOUNG groupAD.CAD31 groupAD.CTRL
## 1          1             0                 0              0              0
## 2          1             0                 0              0              0
## 3          1             0                 0              0              0
## 4          1             1                 0              0              0
## 5          1             1                 0              0              0
## 6          1             1                 0              0              0
## 7          1             0                 1              0              0
## 8          1             0                 1              0              0
## 9          1             0                 1              0              0
## 10         1             0                 1              0              0
## 11         1             0                 0              1              0
## 12         1             0                 0              1              0
## 13         1             0                 0              1              0
## 14         1             0                 0              0              1
## 15         1             0                 0              0              1
## 16         1             0                 0              0              1
## 17         1             0                 0              0              0
## 18         1             0                 0              0              0
## 19         1             0                 0              0              0
## 20         1             0                 0              0              0
## 21         1             0                 0              0              0
## 22         1             0                 0              0              0
```

```

## 23      1      0      0      0
##   groupWT-CAD31 groupWT
## 1      0      0
## 2      0      0
## 3      0      0
## 4      0      0
## 5      0      0
## 6      0      0
## 7      0      0
## 8      0      0
## 9      0      0
## 10     0      0
## 11     0      0
## 12     0      0
## 13     0      0
## 14     0      0
## 15     0      0
## 16     0      0
## 17     1      0
## 18     1      0
## 19     1      0
## 20     0      1
## 21     0      1
## 22     0      1
## 23     0      1
## attr(),"assign")
## [1] 0 1 1 1 1 1 1
## attr(),"contrasts")
## attr(),"contrasts")$group
## [1] "contr.treatment"
data <- data -1

## converting counts to integer mode

## Note: levels of factors in the design contain characters other than
## letters, numbers, '_' and '.'. It is recommended (but not required) to use
## only letters, numbers, and delimiters '_' or '.', as these are safe characters
## for column names in R. [This is a message, not a warning or an error]

## estimating size factors

## Note: levels of factors in the design contain characters other than
## letters, numbers, '_' and '.'. It is recommended (but not required) to use
## only letters, numbers, and delimiters '_' or '.', as these are safe characters
## for column names in R. [This is a message, not a warning or an error]

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## Note: levels of factors in the design contain characters other than
## letters, numbers, '_' and '.'. It is recommended (but not required) to use
## only letters, numbers, and delimiters '_' or '.', as these are safe characters
## for column names in R. [This is a message, not a warning or an error]

## final dispersion estimates

```

```

## Note: levels of factors in the design contain characters other than
## letters, numbers, '_' and '.'. It is recommended (but not required) to use
## only letters, numbers, and delimiters '_' or '.', as these are safe characters
## for column names in R. [This is a message, not a warning or an error]

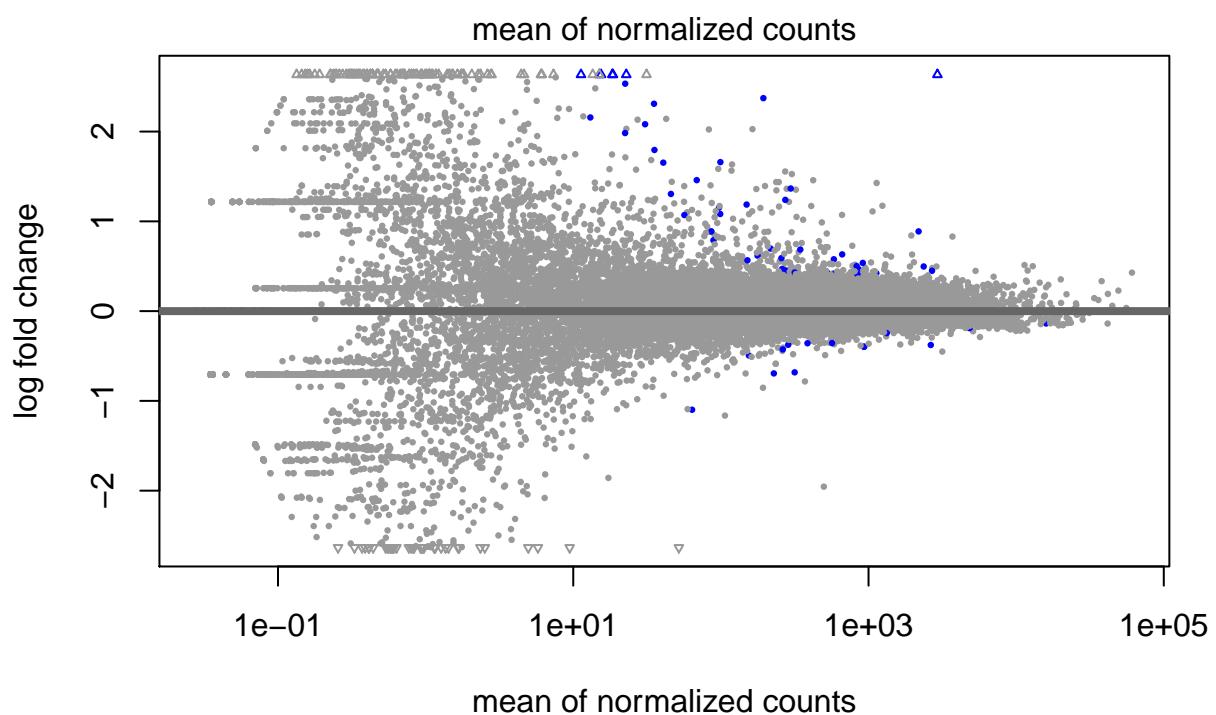
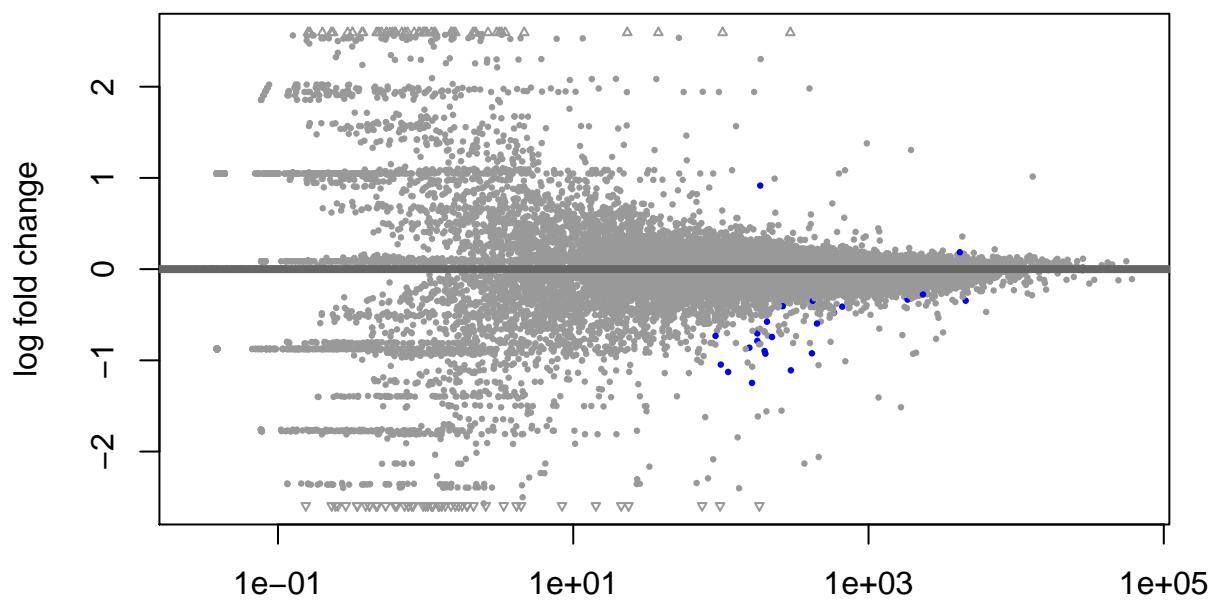
## fitting model and testing

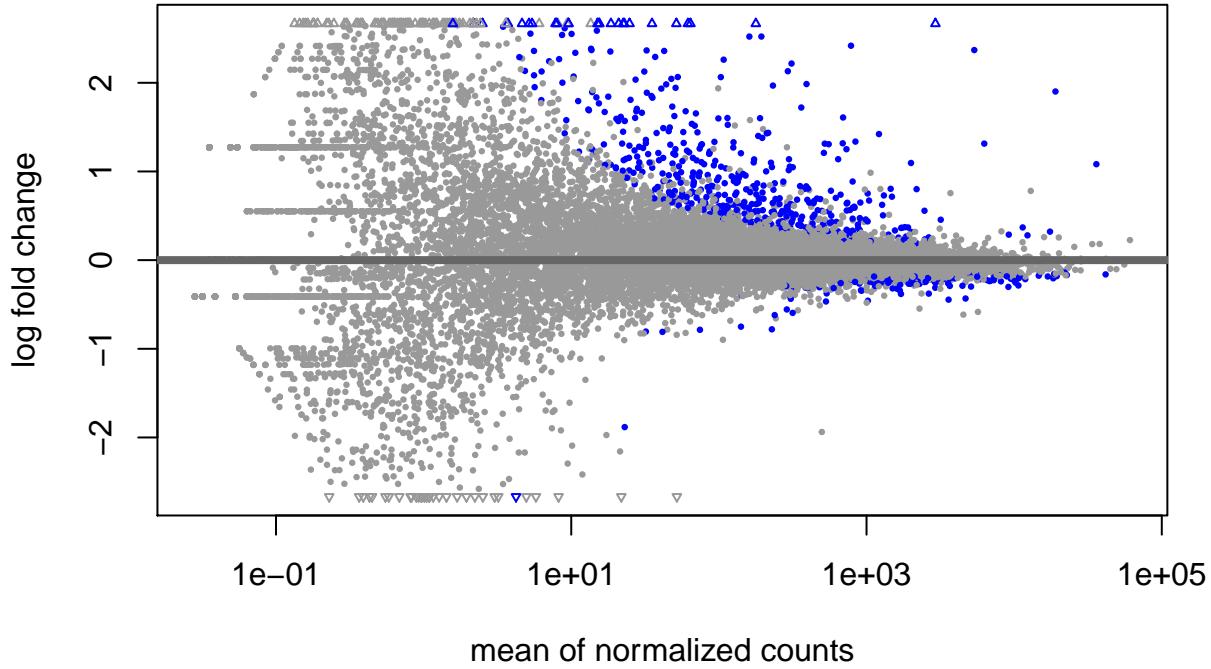
##
## out of 19885 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)      : 2, 0.01%
## LFC < 0 (down)    : 22, 0.11%
## outliers [1]       : 2, 0.01%
## low counts [2]     : 7253, 36%
## (mean count < 35)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
##
## NULL

##
## out of 19885 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)      : 55, 0.28%
## LFC < 0 (down)    : 14, 0.07%
## outliers [1]       : 2, 0.01%
## low counts [2]     : 3818, 19%
## (mean count < 2)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
##
## NULL

##
## out of 19885 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)      : 487, 2.4%
## LFC < 0 (down)    : 148, 0.74%
## outliers [1]       : 2, 0.01%
## low counts [2]     : 3436, 17%
## (mean count < 2)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
##
## NULL

```





```

#function
get_deg <- function(data, group, design){
  dl <- DGEList(as.matrix(data), lib.size = colSums(data), norm.factors = rep(1,ncol(data)), group = group)
  dl <- calcNormFactors(dl)
  plotMDS(dl)

  ed <- estimateDisp(dl, design = design)
  plotBCV(ed)

  et <- exactTest(ed)

  toptag<- topTags(et,n = 1000, p.value = 0.05)
  return(toptag)
}

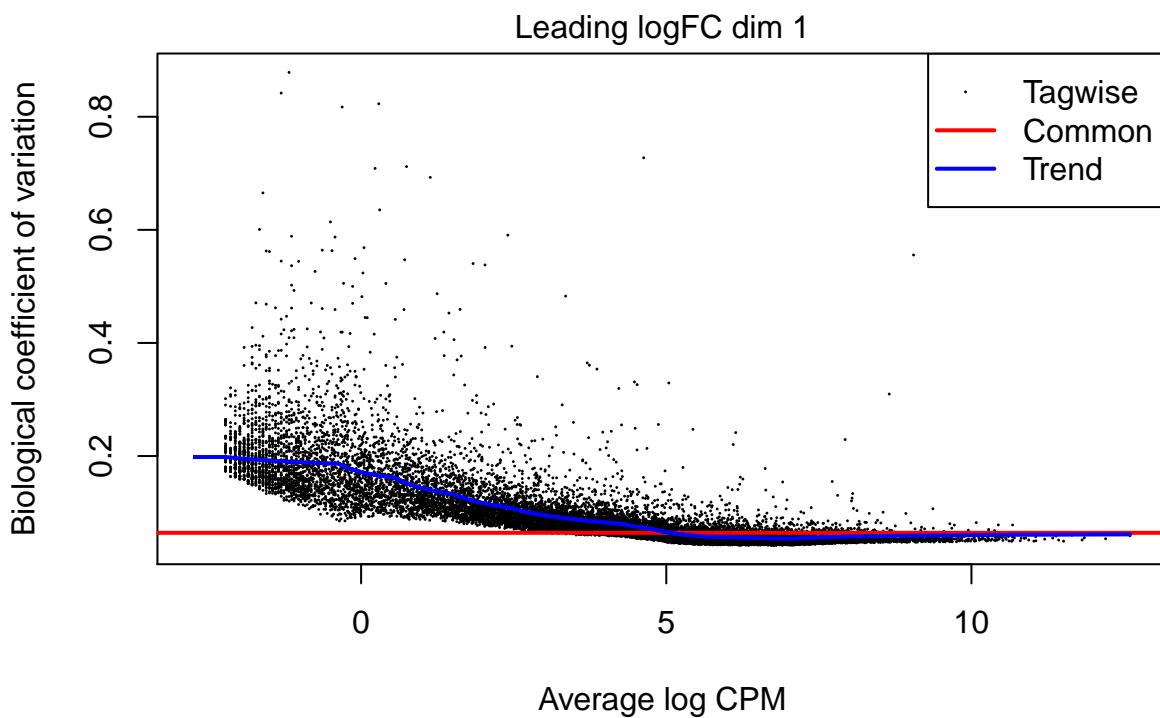
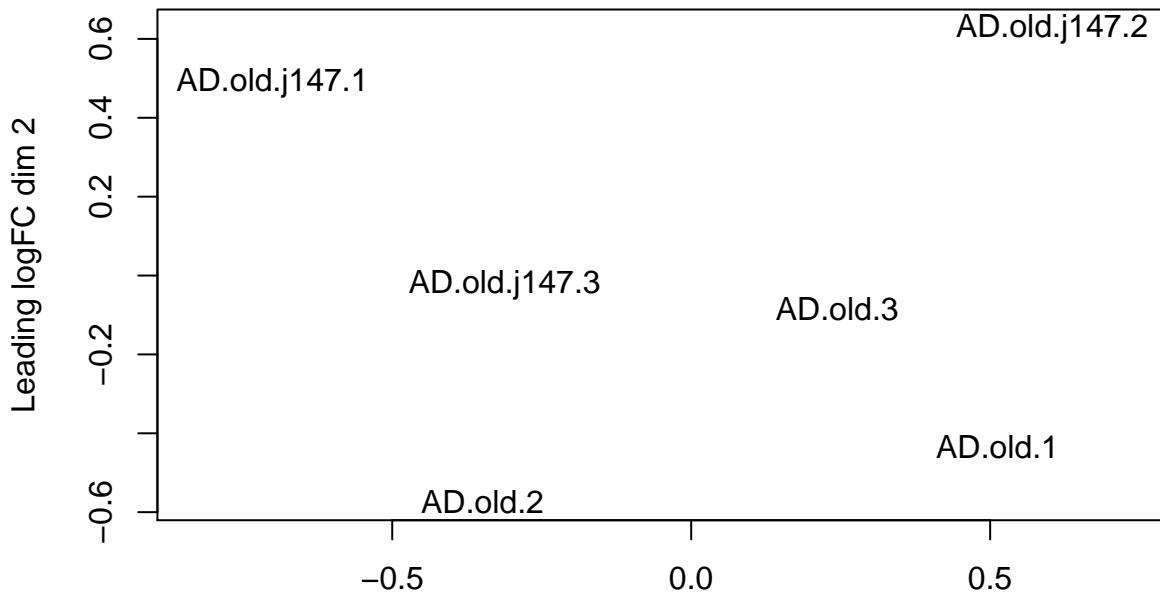
# Make the comparison data
J147.data <- data[1:6]
J147.group <- factor(c(1,1,1,rep(2,each=3)),
                      labels = c("AD.J147", "AD.CTRL.OLD"))
J147.design <- model.matrix(~ J147.group)

cad31.AD.cad31.data <- data[11:16]
cad31.AD.cad31.group <- factor(c(1,1,1,rep(2,each=3)),
                                 labels = c("AD.CAD31", "AD"))
cad31.AD.cad31.design <- model.matrix(~ cad31.AD.cad31.group)

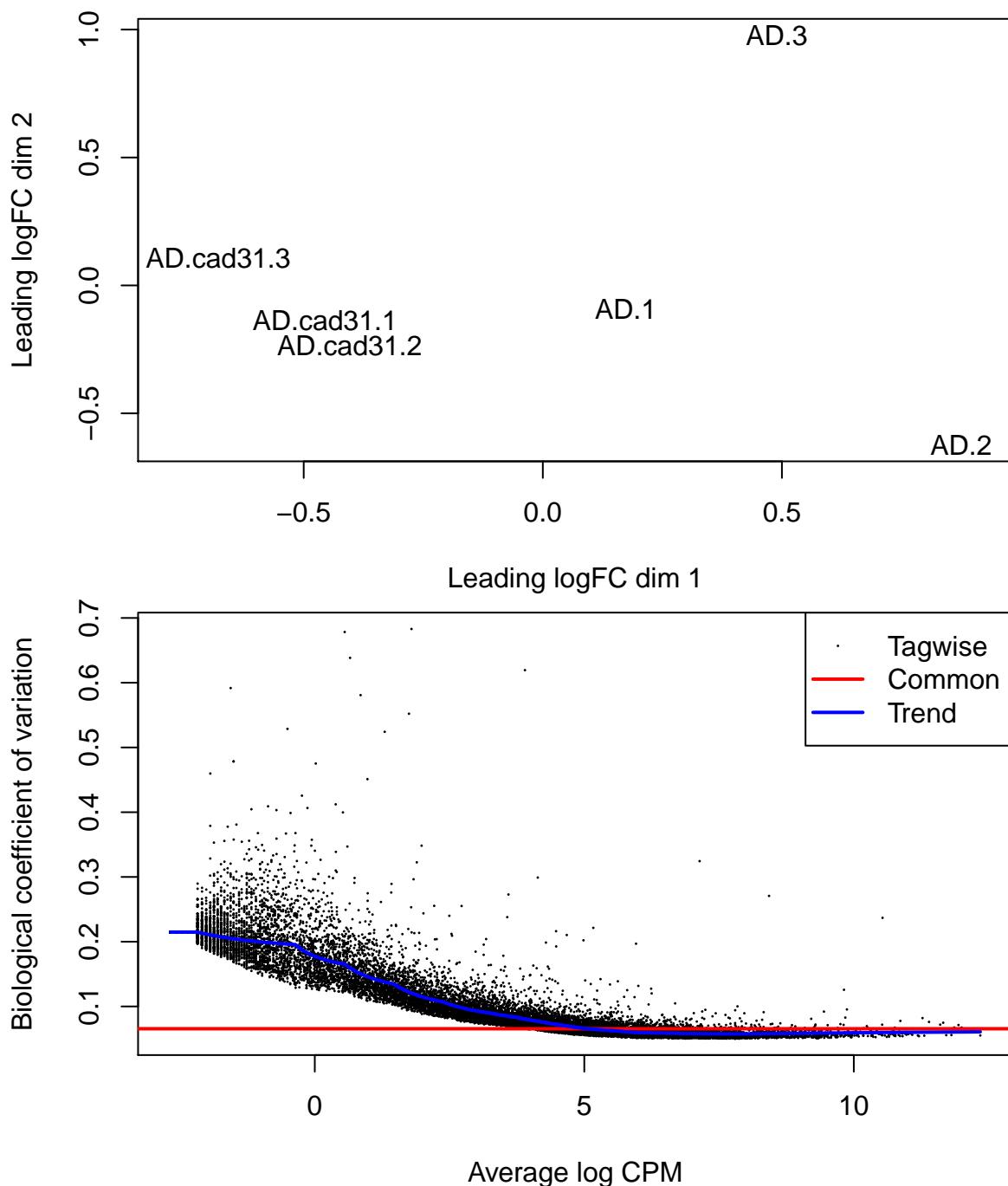
cad31.wt.cad31.data <- data[17:23]
cad31.wt.cad31.group <- factor(c(1,1,1,rep(2,each=4)),
                                labels = c("WT.CAD31", "WT"))
cad31.wt.cad31.design <- model.matrix(~ cad31.wt.cad31.group)

```

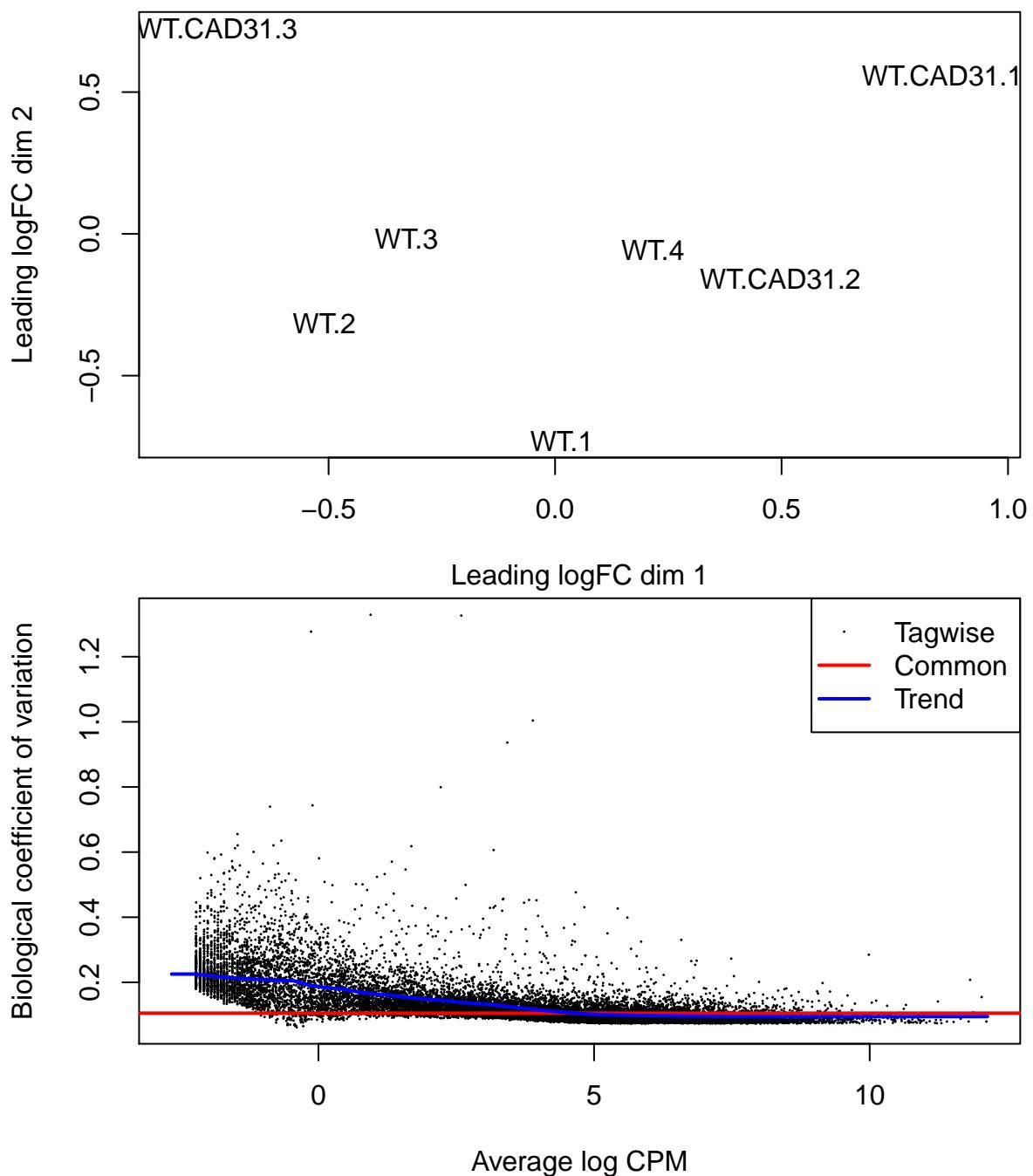
```
#do the analysis  
j147.toptag <- get_deg(J147.data, J147.group, J147.design)
```



```
cad31.AD.cad31.toptag <- get_deg(cad31.AD.cad31.data, cad31.AD.cad31.group, cad31.AD.cad31.design)
```



```
cad31.wt.cad31.toptag <- get_deg(cad31.wt.cad31.data, cad31.wt.cad31.group, cad31.wt.cad31.design)
```

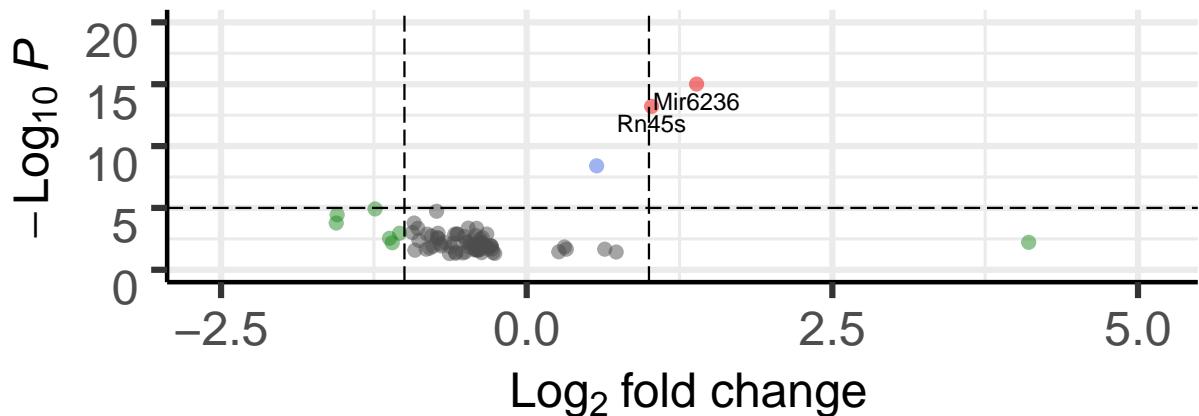


```
EnhancedVolcano(j147.toptag$table, x = 'logFC', y = 'FDR', lab = rownames(j147.toptag$table))
```

## Volcano plot

EnhancedVolcano

● NS ● Log<sub>2</sub> FC ● p-value ● p – value and log<sub>2</sub> FC



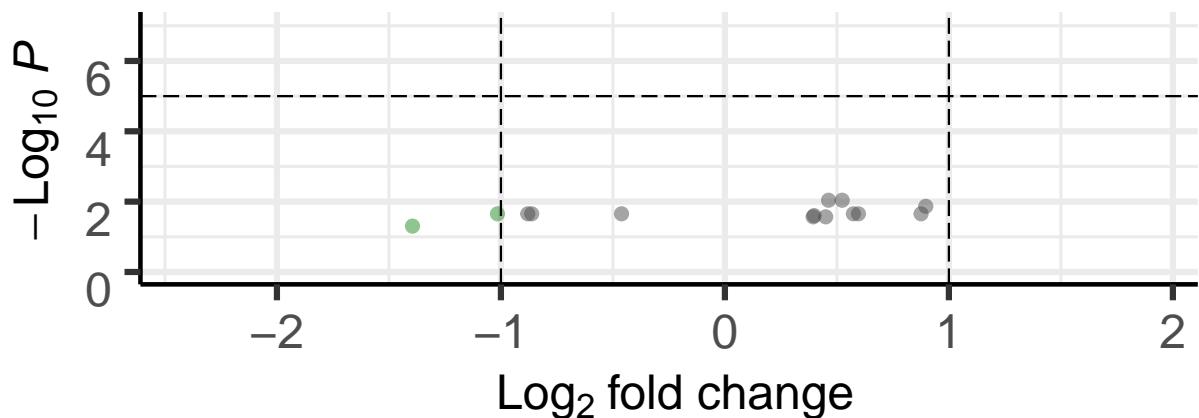
Total = 76 variables

```
EnhancedVolcano(cad31.wt.cad31.toptag$table, x = 'logFC', y = 'FDR', lab = rownames(cad31.wt.cad31.toptag$
```

## Volcano plot

EnhancedVolcano

● NS ● Log<sub>2</sub> FC ● p-value ● p – value and log<sub>2</sub> FC



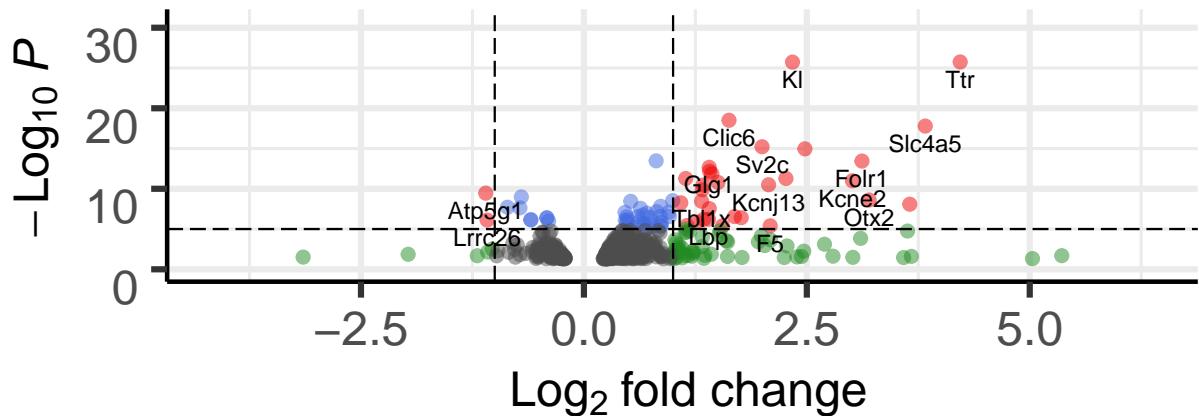
Total = 14 variables

```
EnhancedVolcano(cad31.AD.cad31.toptag$table, x = 'logFC', y = 'FDR', lab = rownames(cad31.AD.cad31.toptag$table))
```

## Volcano plot

## EnhancedVolcano

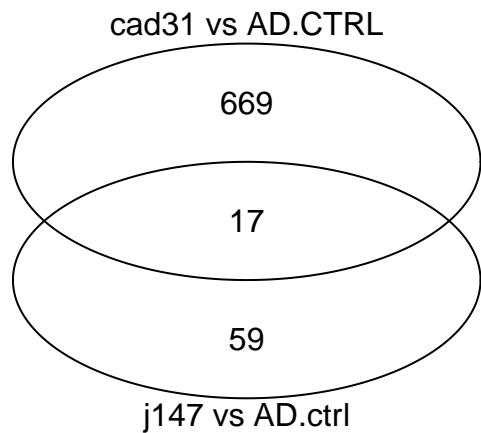
● NS ● Log<sub>2</sub> FC ● p-value ● p – value and log<sub>2</sub> FC



Total = 686 variables

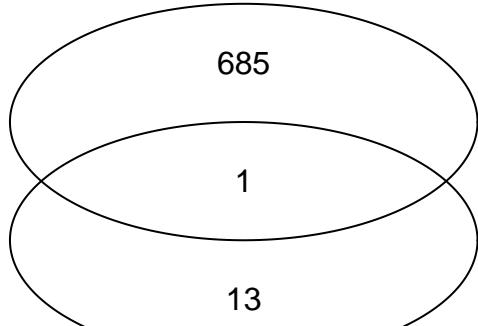
# Create a Venn-diagram given just the list of gene-names for both sets

```
venn(list("j147 vs AD.ctrl" = rownames(j147.toptag),
          "cad31 vs AD CTRL" = rownames(cad31.AD.cad31.toptag)), )
```



```
venn(list("WT vs CAD31.WT" = rownames(cad31.wt.cad31.toptag),
          "cad31 vs AD CTRL" = rownames(cad31.AD.cad31.toptag)), )
```

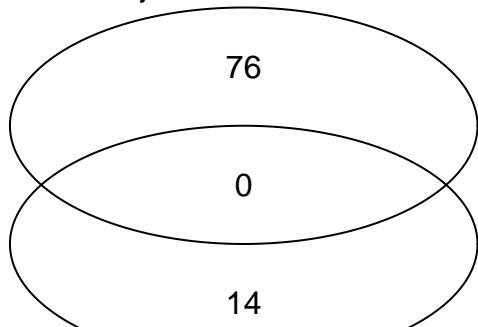
cad31 vs AD.CTRL



WT vs CAD31.WT

```
venn(list("WT vs CAD31.WT" = rownames(cad31.wt.cad31.toptag),
          "j147 vs AD.ctrl" = rownames(j147.toptag)), )
```

j147 vs AD.ctrl



WT vs CAD31.WT

```
degs <- c(rownames(j147.toptag) , rownames(cad31.wt.cad31.toptag) , rownames(cad31.AD.cad31.toptag))

# Write gene names to a file
write.table(degs, file = "edger-deg-names.txt",
            row.names = FALSE, quote = FALSE, col.names = FALSE)
```