Project 2 Group 4

von Anna, Ann-Sophie und Jana

Load data

```
NCI_TPW_gep_treated <- readRDS(url("https://ndownloader.figshare.com/files/14720180?private_link=db1411
NCI_TPW_gep_untreated <- readRDS(url("https://ndownloader.figshare.com/files/14720183?private_link=db14
NCI_TPW_metadata <- read.delim("https://ndownloader.figshare.com/files/14720186?private_link=db1411debc
NegLogGI50 <- readRDS(url("https://ndownloader.figshare.com/files/ 14720210?private_link=074e0120fe5e68
CCLE_basalexpression <- readRDS(url("https://ndownloader.figshare.com/files/14770127?private_link=fc0c7
CCLE_copynumber <- readRDS(url("https://ndownloader.figshare.com/files/14770130?private_link=fc0c71246d
CCLE_mutations <- readRDS(url("https://ndownloader.figshare.com/files/14770133?private_link=fc0c71246dc
cellline_annotation <-read.delim("https://ndownloader.figshare.com/files/14768981?private_link=efb6a529
drug_annotation <- read.delim("https://ndownloader.figshare.com/files/14768984?private_link=efb6a529eaf
```

1. Broad analysis

Data preparation and annotation

Calculate fold change due to drug treatment

```
fold_changes <- NCI_TPW_gep_treated - NCI_TPW_gep_untreated
fold_changes <- as.data.frame(fold_changes)
```

Renaming of cellline SK-MEL-2 Problem: name of cellline SK-MEL-2 is part of cellline SK-MEL_28 Solution: rename SK-MEL-2 to SK-MEL-2_ (first define it as new factor level)

```
levels(cellline_annotation$Cell_Line_Name) <- c(levels(cellline_annotation$Cell_Line_Name),
                                                "SK-MEL-2_")
cellline_annotation[33, 1] <- "SK-MEL-2_"
#delete level SK-MEL-2 (otherwise we would have 62, instead of 61 levels)
cellline_annotation$Cell_Line_Name <- factor(as.character(
  cellline_annotation$Cell_Line_Name))
```

Create annotation: A matrix is created, which contains for each sample name the drug, cellline and cancertype

1. Drug

```
sample_drug <- as.data.frame(sapply(levels(drug_annotation$Drug), grepl,
                                    colnames(fold_changes), ignore.case = TRUE))
  #creates table with TRUE and FALSE for each sample and drug
rownames(sample_drug) <- colnames(fold_changes)
drugs <- as.vector(apply(sample_drug, 1, function(x){
  colnames(sample_drug[which(x)])
}))
```

2. Cellline

```
sample_cellline <- as.data.frame(sapply(levels(cellline_annotation$Cell_Line_Name), grepl,
                                        colnames(fold_changes), ignore.case = TRUE))
  #creates table with TRUE and FALSE for each sample and cellline
rownames(sample_cellline) <- colnames(fold_changes)
cellline <- as.vector(unlist(apply(sample_cellline, 1, function(x){
  colnames(sample_cellline[which(x)])
})))

annotation <- cbind("Drug" = drugs, "Cellline" = cellline)
rownames(annotation) <- colnames(fold_changes)
```

3. Cancertype

```
cancertype <- sapply(annotation[, 2], function(x){
  #2nd column contains cellline annotation of samples
  cellline_annotation$Cancer_type[cellline_annotation$Cell_Line_Name == x]
})
cancertype <- as.vector(unlist(cancertype))

annotation <- cbind(annotation, "Cancertype" = cancertype)
rm(drugs, sample_drug, cellline, sample_cellline, cancertype)
```

Coloring:
Create a vector which assigns each drug or each cancertype a color

1. According to drug (color_vector_all_drugs)

```
#define a color palette with 15 chosen colors
color_palette_drug <- c("aquamarine", "brown", "forestgreen", "slategrey",
                        "chartreuse", "darkgoldenrod1", "cadetblue","purple",
                        "firebrick1", "deepskyblue", "gold", "violetred4",
                        "deeppink", "plum2", "blue" )
names(color_palette_drug) <- levels(drug_annotation$Drug)

#create vector containing a color name for each sample according to drug
color_vector_drug <- sapply(rownames(annotation), function(x){
  unname(color_palette_drug[annotation[x, 1]]) #first column of annotation contains drug
})
```

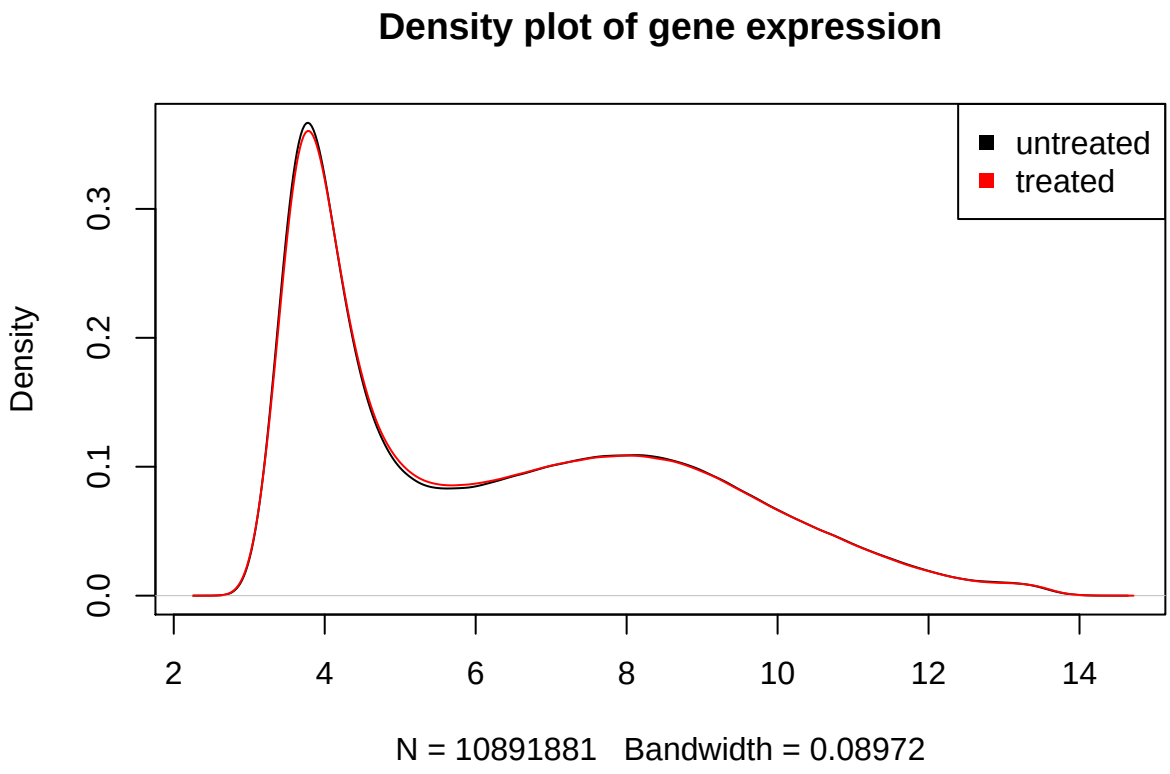
2. According to cancertype (color_vector_cancertype)

```
#define a color palette with 9 chosen colors
color_palette_cancertype <- c("aquamarine", "brown", "forestgreen", "chartreuse",
                              "darkgoldenrod1", "cadetblue","purple",
                              "firebrick1", "deepskyblue")
names(color_palette_cancertype) <- levels(cellline_annotation$Cancer_type)

#create vector containing a color name for each sample according to cancertype
color_vector_cancertype <- sapply(rownames(annotation), function(x){
  unname(color_palette_cancertype[annotation[x, 3]]) #3rd columns of annotation contains cancertype
})
```

Density plot

To show the distribution of all gene expression values of all samples, a density plot was drawn. The black line contains all values measured for control samples (untreated). In red the distribution of the gene expressiion of all samples treated with 15 drugs is shown.

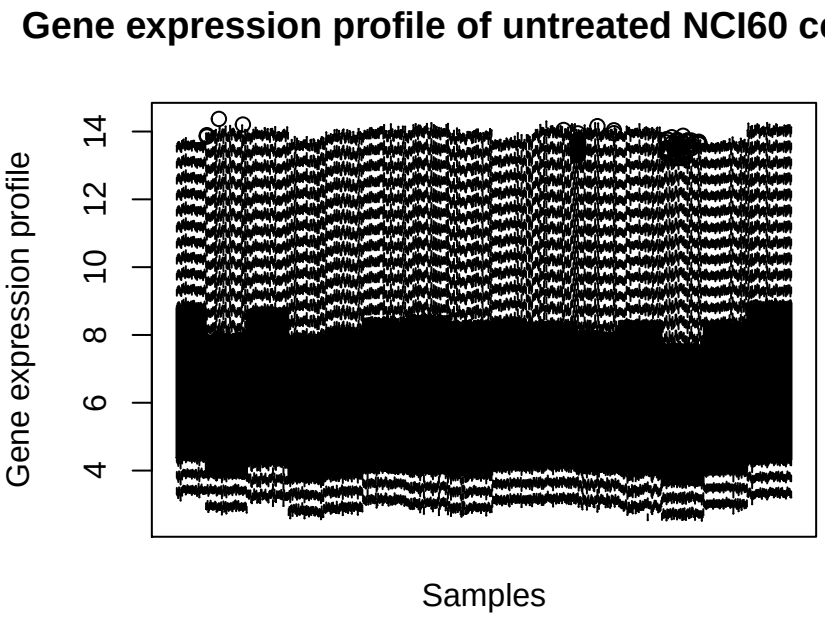
```
plot(density(NCI_TPW_gep_untreated), "Density plot of gene expression")
lines(density(NCI_TPW_gep_treated), col = "red")
legend("topright", legend = c("untreated", "treated"), col = c("black", "red"), pch = 15)
```



Boxplot

Create a boxplot to show the distribution of the foldchanges of all genes in one box per sample

```
#par makes spaces outside the plot larger, xaxt: removes labels on x-axis
#title() used to move xlab nearer to the axis
par(oma = c(1, 1, 1, 8), xpd = "TRUE")
boxplot(NCI_TPW_gep_untreated,
        xaxt = "n",
        ylab = "Gene expression profile",
        vertical =  T,
        main = "Gene expression profile of untreated NCI60 celllines")
title(xlab = "Samples", line = 1.0)
```



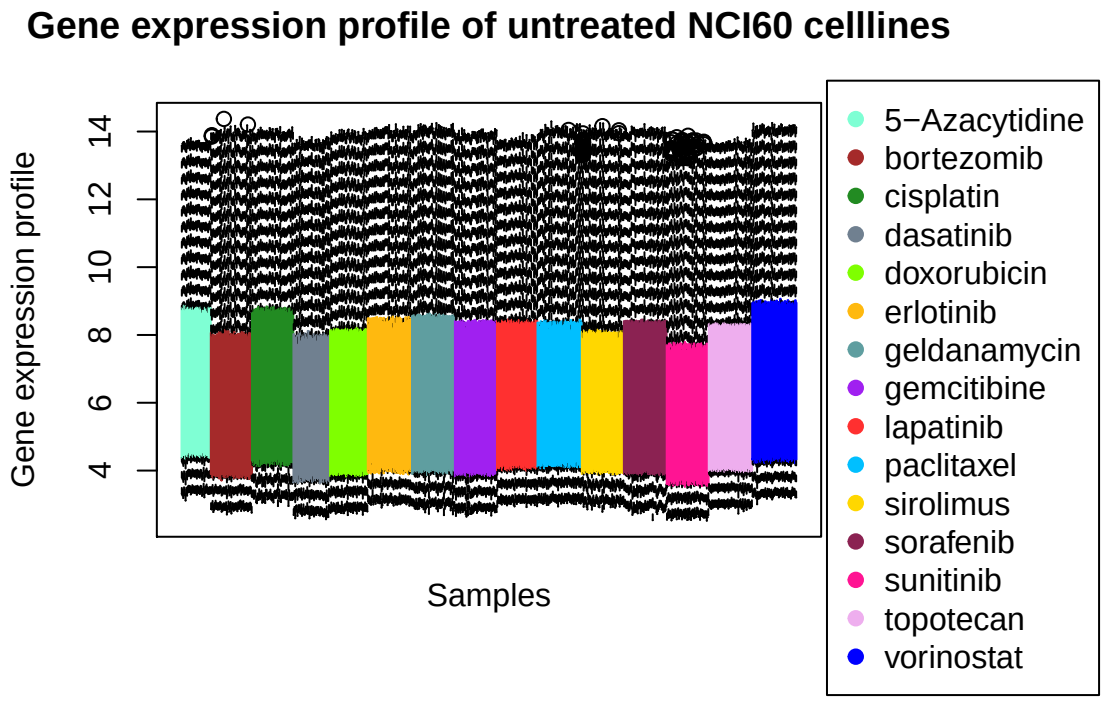
Batch effect was seen –> corresponding to drugs?

Color plot according to drugs

```
par(oma = c(1, 1, 1, 8), xpd = "TRUE")
```

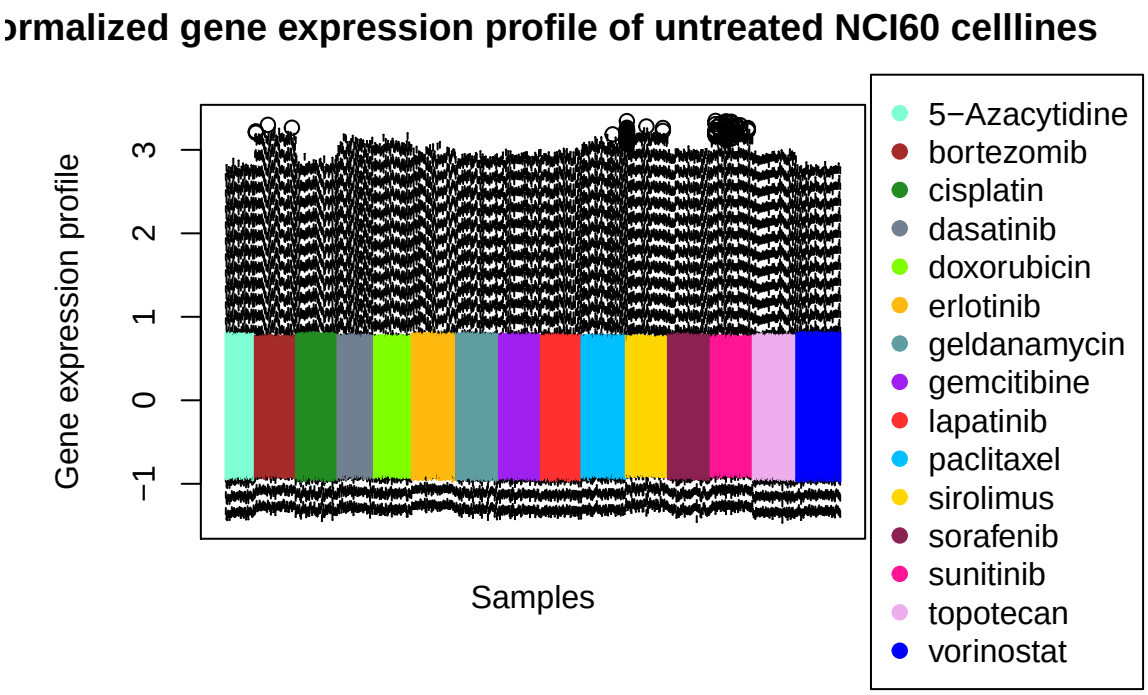
```
## Warning in par(oma = c(1, 1, 1, 8), xpd = "TRUE"): NAs durch Umwandlung
## erzeugt
```

```
boxplot(NCI_TPW_gep_untreated,
        xaxt = "n",
        ylab = "Gene expression profile",
        vertical =  T,
        main = "Gene expression profile of untreated NCI60 celllines",
        boxcol = color_vector_drug)
title(xlab = "Samples", line = 1.0)
legend(x = 860,
       y = 15.5,
       legend = names(color_palette_drug),
       col = color_palette_drug,
       pch = 19)
```



Normalization of data is necessary

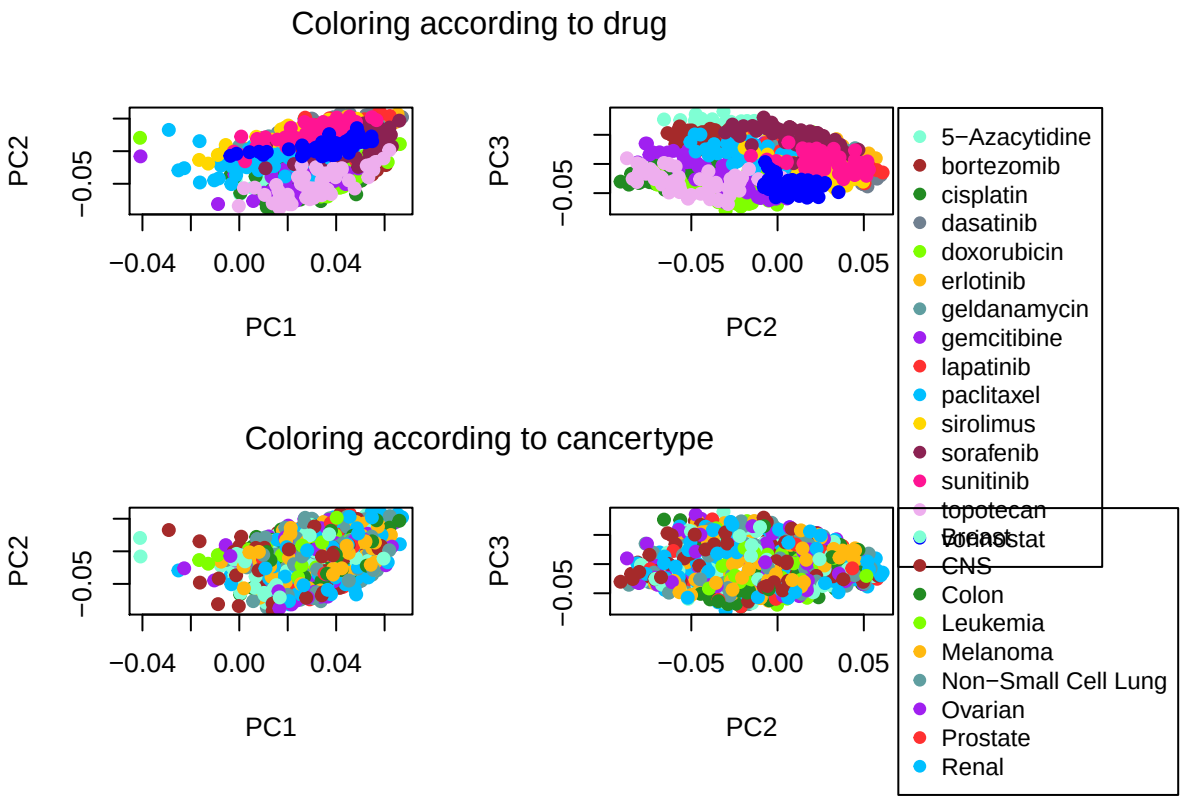
```
#each sample should have mean 0 and sd 1
untreated_normalized <- apply(NCI_TPW_gep_untreated, 2, function(x){
  (x - mean(x)) / sd(x)
})
FC_normalized <- apply(fold_changes, 2, function(x){
  (x - mean(x)) / sd(x)
})

#boxplot of normalized untreated values
par(oma = c(1, 1, 1, 8), xpd = "TRUE")
boxplot(untreated_normalized,
        xaxt = "n",
        ylab = "Gene expression profile",
        vertical =  T,
        main = "Normalized gene expression profile of untreated NCI60 celllines",
        boxcol = color_vector_drug)
title(xlab = "Samples", line = 1.0)
legend(x = 860,
       y = 3.9,
       legend = names(color_palette_drug),
       col = color_palette_drug,
       pch = 19)
```



PCA

```
pca <- prcomp(FC_normalized)

#color PCA according to drug
par(oma = c(1, 1, 1, 8), mfrow = c(2, 2)) #mfrow to create multiple plots
#PC1 and PC2
plot(pca$rotation[,1],
     pca$rotation[,2],
     col = color_vector_drug,
     pch = 19,
     xlab = "PC1",
     ylab = "PC2")
#PC2 and PC3
plot(pca$rotation[,2],
     pca$rotation[,3],
     col = color_vector_drug,
     pch = 19, xlab = "PC2",
     ylab = "PC3")
#create legend on the right side
legend(x = 0.07,
       y = 0.096,
       legend = names(color_palette_drug),
       col = color_palette_drug,
```

```
       pch = 19,
       xpd = "TRUE",
       cex = 0.9)
#Title: mtext = margin text, side = 3 (upside)
mtext("Coloring according to drug", side = 3, line = -2, outer = TRUE)

#Color PCA according to cancertype
#PC1 and PC2
plot(pca$rotation[,1],
     pca$rotation[,2],
     col = color_vector_cancertype,
     pch = 19, xlab = "PC1",
     ylab = "PC2")
#PC2 and PC3
plot(pca$rotation[,2],
     pca$rotation[,3],
     col = color_vector_cancertype,
     pch = 19, xlab = "PC2",
     ylab = "PC3")
legend(x = 0.07,
       y = 0.096,
       legend = names(color_palette_cancertype),
       col = color_palette_cancertype,
       pch = 19,
       xpd = "TRUE",
       cex = 0.9)
mtext("Coloring according to cancertype", side = 3, line = -15, outer = TRUE)
```

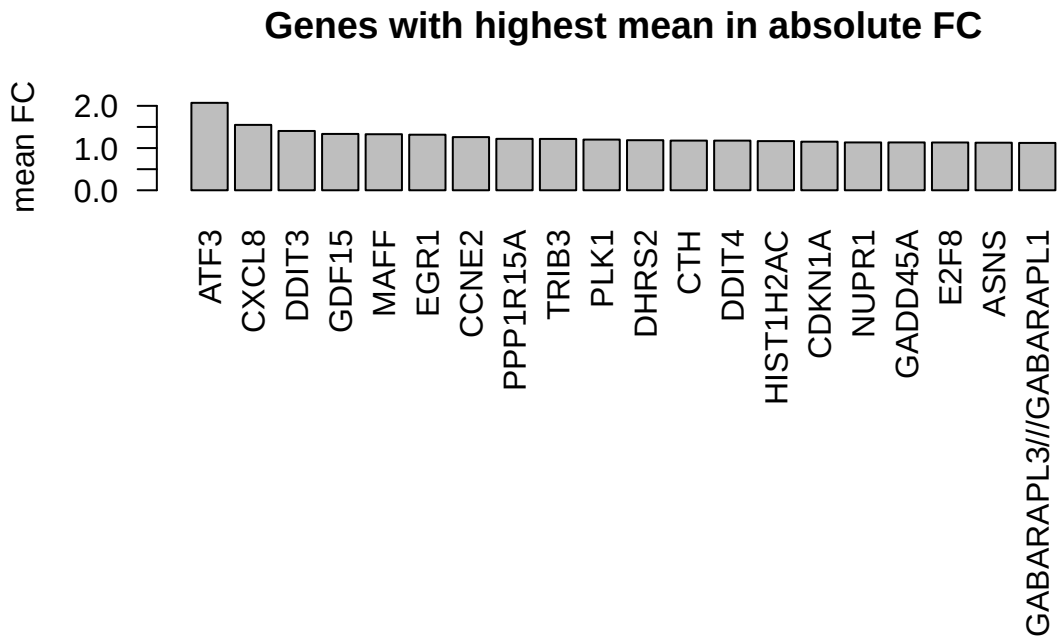



```
rm(pca)
```

Most regulated genes

Barplot to find genes, which were mostly regulated by all treatments

```
#calculating the mean FC over positive FC values
mean_FC_abs <- apply(abs(fold_changes), 1, mean)
mean_FC_abs <- sort(mean_FC_abs, decreasing = TRUE)
par(oma = c(10, 1, 1, 1))
barplot(mean_FC_abs[1:20],
        main = "Genes with highest mean in absolute FC",
        ylab = "mean FC",
        las = 2)
```



boxplot of genes with highest mean FC

```
FC_samples_with_highest_mean_FC <- as.data.frame(sapply(names(mean_FC_abs)[1:20], function(x){
  fold_changes[which(x == rownames(fold_changes)),]
}))
# boxplot(FC_samples_with_highest_mean_FC,
      '  ylab = "foldchange",
        main = "boxplot of foldchange of the genes with highest mean FC",
        las=2) '
```

```
## [1] "  ylab = \"foldchange\", \n        main = \"boxplot of foldchange of the genes with highest mean
```

Specific analysis: Erlotinib

2. Milestone: find most affected cell lines and genes

Data preparation

Erlotinib treated cell lines are selected and the matrix of the foldchange is normalized

```
#new matrix only with samples/columns treated with erlotinib  (e=erlotinib)
e_treated <- NCI_TPW_gep_treated[,grep ("erlotinib", colnames(NCI_TPW_gep_treated))]
e_untreated <- NCI_TPW_gep_untreated[,grep ("erlotinib", colnames(NCI_TPW_gep_treated))]
e_foldchange <- e_treated - e_untreated
```

```
#colnames of e_foldchange with cellline instead of complete sample name
cellline <- sapply(colnames(e_foldchange), function(x){
  annotation[x,"Cellline"]
  })
colnames(e_foldchange) <- cellline

#e_foldchange_normalized: z-Transformation to get mean=0 and sd=1
e_foldchange_normalized <- apply(e_foldchange, 2, function(x){
  (x - mean(x)) / sd(x)
})
```

Most regulated cell lines

Table of 15 cell lines

Cell lines, which showed the highest variance over all genes were selected

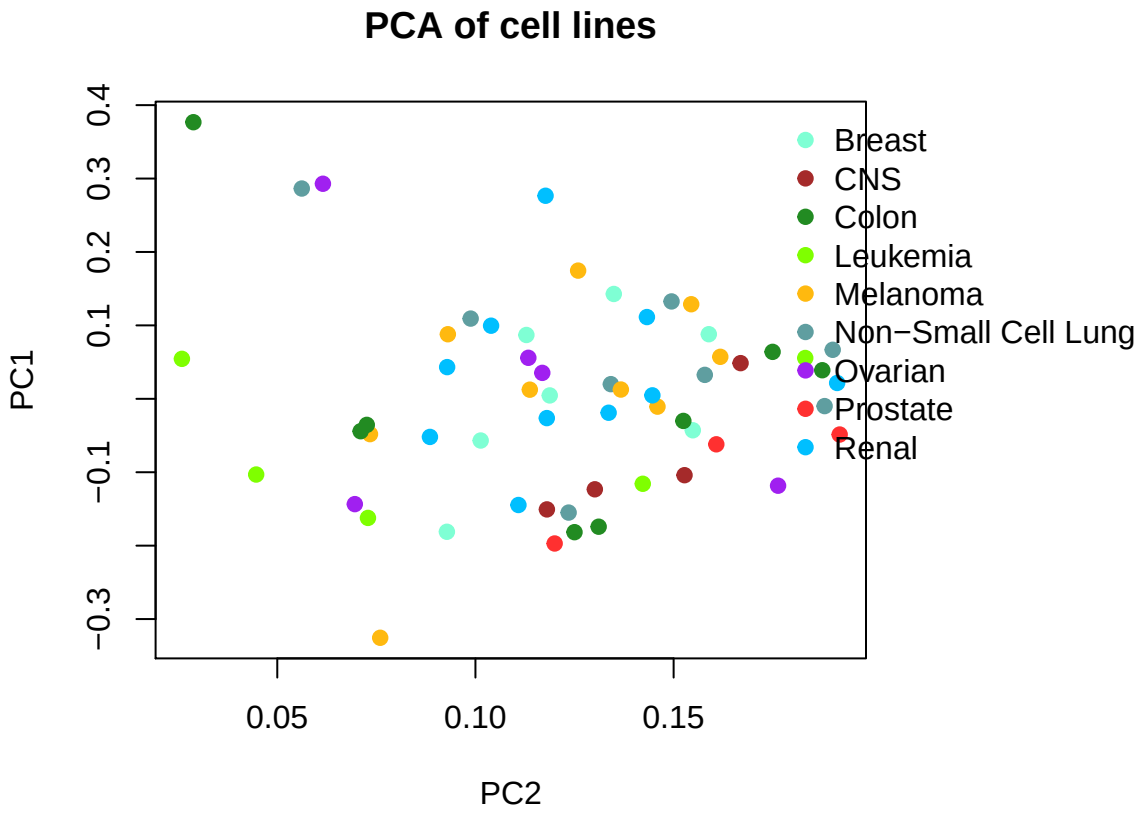
```
#select 15 cell lines with highest variance (greater than 75% quantile, sorted by decreasing value)
var_cell_line <- apply(e_foldchange, 2, var)
cell_line_var_greater_75quantile <- sort(var_cell_line [which (abs(var_cell_line) > quantile(abs(var_ce
cell_line_var_greater_75quantile <- round(cell_line_var_greater_75quantile, digits=5)

#add column with cell line for top15 celllines
celllines_top15 <- as.data.frame(names(cell_line_var_greater_75quantile))

#add column with cancertype for top15 celllines
annotation_cancertype <- annotation[,"Cancertype"]
names(annotation_cancertype) <- colnames(e_foldchange)
cancertypes_top15 <- sapply(names(cell_line_var_greater_75quantile), function(x) {annotation_cancertype
})
table_cell_lines_var_top15 <- cbind(celllines_top15, cell_line_var_greater_75quantile, cancertypes_top1

colnames(table_cell_lines_var_top15) <- c("Cellline", "Variance", "Cancertype")
rownames(table_cell_lines_var_top15) <- c(1:nrow(celllines_top15))
print(table_cell_lines_var_top15)
```

```
     Cellline Variance          Cancertype
1   NCI-H322M  0.45385               Renal
2        ACHN  0.34745               Renal
3     IGR-OV1  0.33955             Leukemia
4     SK-OV-3  0.29909 Non-Small Cell Lung
5      CAKI-1  0.20350             Prostate
6     OVCAR-3  0.20219               Renal
7       HL-60  0.19209             Melanoma
8    CCRF-CEM  0.18592               Colon
9  MDA-MB-468  0.17965                 CNS
10      SN12C  0.16687               Colon
11   NCI-H522  0.16137              Breast
12      K-562  0.14258 Non-Small Cell Lung
13     HCT-15  0.13979               Colon
14     DU-145  0.13966 Non-Small Cell Lung
15         SR  0.13955 Non-Small Cell Lung
```

PCA

PCA is performed to find cell lines, which differ most from the other cell lines

```
#PCA with transformed matrix (each point represents a sample):
par(mar= c(4,4,4,10))
pca <- prcomp(e_foldchange_normalized)
plot(pca$rotation[,1], pca$rotation[,3], col=color_vector_cancertype, pch=19, xlab = "PC2", ylab="PC1",
legend("topright", legend= names(color_palette_cancertype), col= color_palette_cancertype, pch=19, xpd="
```



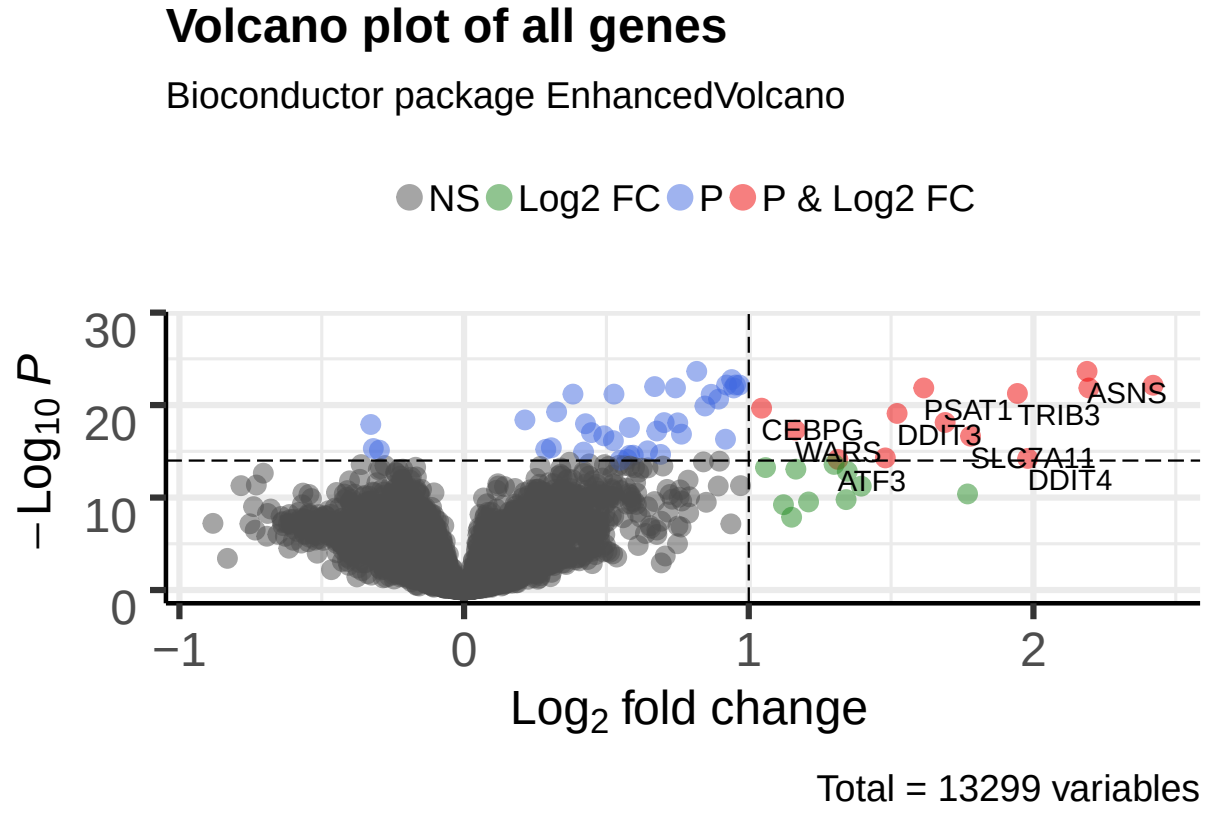
Most regulated genes

Volcano plot

Create volcano plot to find the genes with the highest fold change and highest significance

```
#mean of gene expression of each gene over all cell lines
e_foldchange_mean_over_cell_lines <- rowMeans(e_foldchange) #equal to e_treated_mean_over_cell_lines -

#determine the p-value for a paired two-sample t-test
p_values <- sapply(rownames(e_treated), function(x) {
  t.test(e_treated[x,], e_untreated[x,],paired= T)$p.value}) # perform t-test and save p-values of each
FDR_values <- p.adjust(p_values, method = "BH", n = length(p_values))#calculate FDR with benjamini-hoch
```

```
#table of results
statistics_values <- cbind(e_foldchange_mean_over_cell_lines, FDR_values)
#coloring with package enhanced volcano
#install package EnhancedVolcano (needs ggplot2, ggrepel)
library(EnhancedVolcano)

EnhancedVolcano(statistics_values,
                lab = rownames(statistics_values),
                x = "e_foldchange_mean_over_cell_lines", #colname of FC values in this table (statistic
                y = "FDR_values", #colname of FDR (statistics_values)
                title = "Volcano plot of all genes",
                pCutoff = 10e-15, #threshold for coloring significant ones
                FCcutoff = 1, #threshold for coloring high FC
                transcriptPointSize = 3,
                transcriptLabSize = 4.0)
```

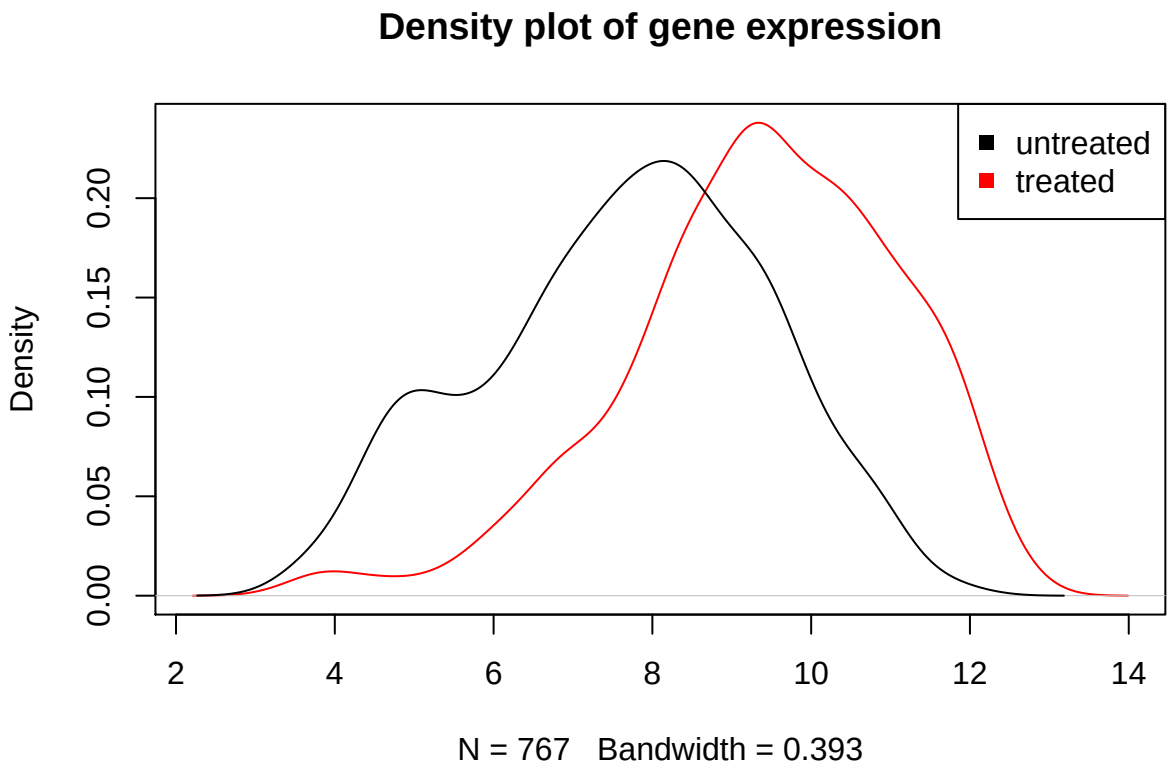


Density plot

Draw a density plot only with biomarkers identified by volcano plot

```
#save the "red" genes seen in the volcano plot in a vector for further analysis
biomarkers <- rownames(statistics_values)[which(abs(statistics_values[, 1]) > 1
                                                & statistics_values[, 2] < 10e-15)]
```

```r
#Density plot with these genes (untreated vs. treated)
plot(density(e_treated[biomarkers, ]), "Density plot of gene expression", col = "red")
lines(density(e_untreated[biomarkers, ]), col = "black")
legend("topright", legend = c("untreated", "treated"), col = c("black", "red"), pch = 15)
```

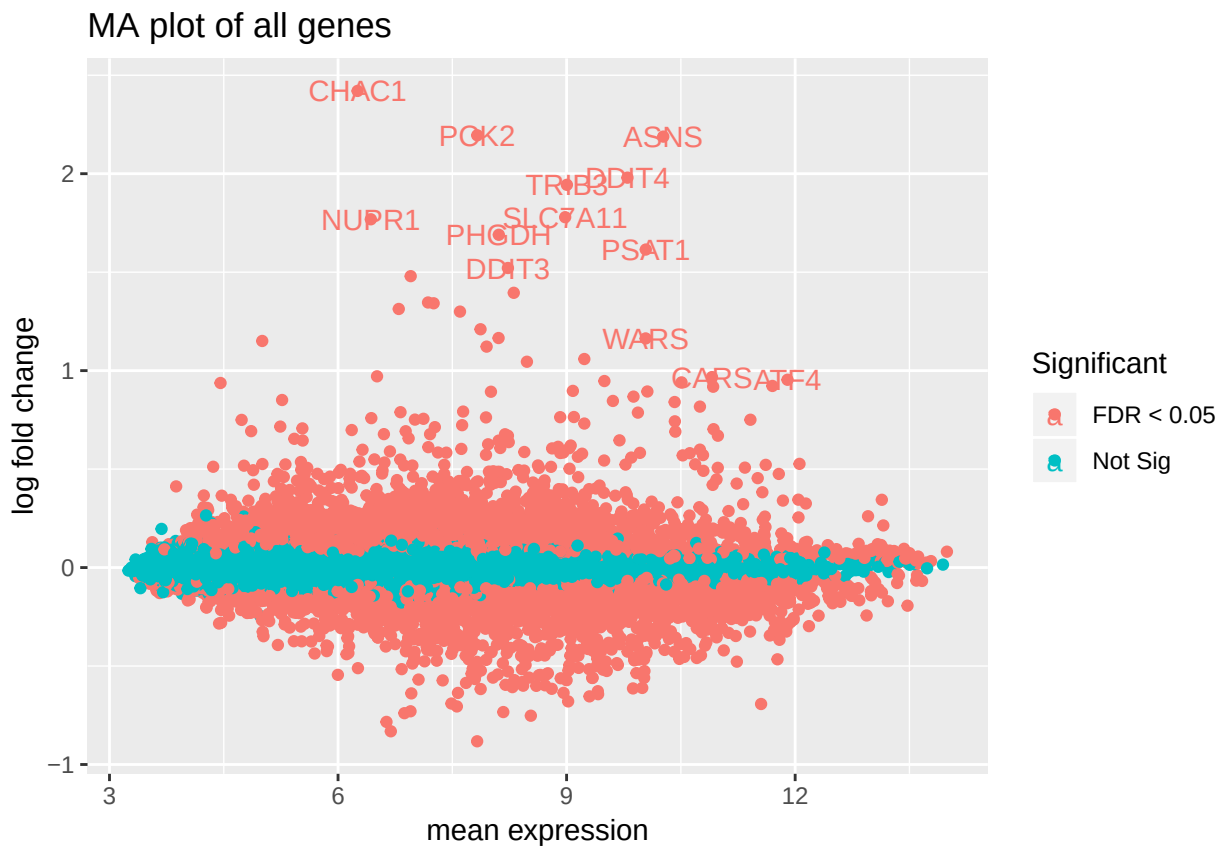


MA-Plot

Draw an MA plot to compare the fold change to the mean expression of all genes

```r
#install package and load ggplot2 and ggrepel
library(ggplot2)
library(ggrepel)

#create matrices with the variables M and A of a MA-plot
M <- e_foldchange # M= log2(treated) - log2 (untreated)
A <- 1/2*(e_treated+ e_untreated) # average log2-expression value A = 1/2 (log2(treated)+log2(untreated
MA <- cbind("M"= rowMeans(M), "A" = rowMeans(A), FDR_values)
rm(M, A)
MA <- as.data.frame(MA)
MA$Significant <- ifelse(MA$FDR_values<0.05, "FDR < 0.05", "Not Sig")

#matrix with important genes of MA plot
MA_labeled <- MA[which(MA[ , "M"] > 1.5 | MA[,"M"] > 0.95 & MA[,"A"] > 10) , ]
```

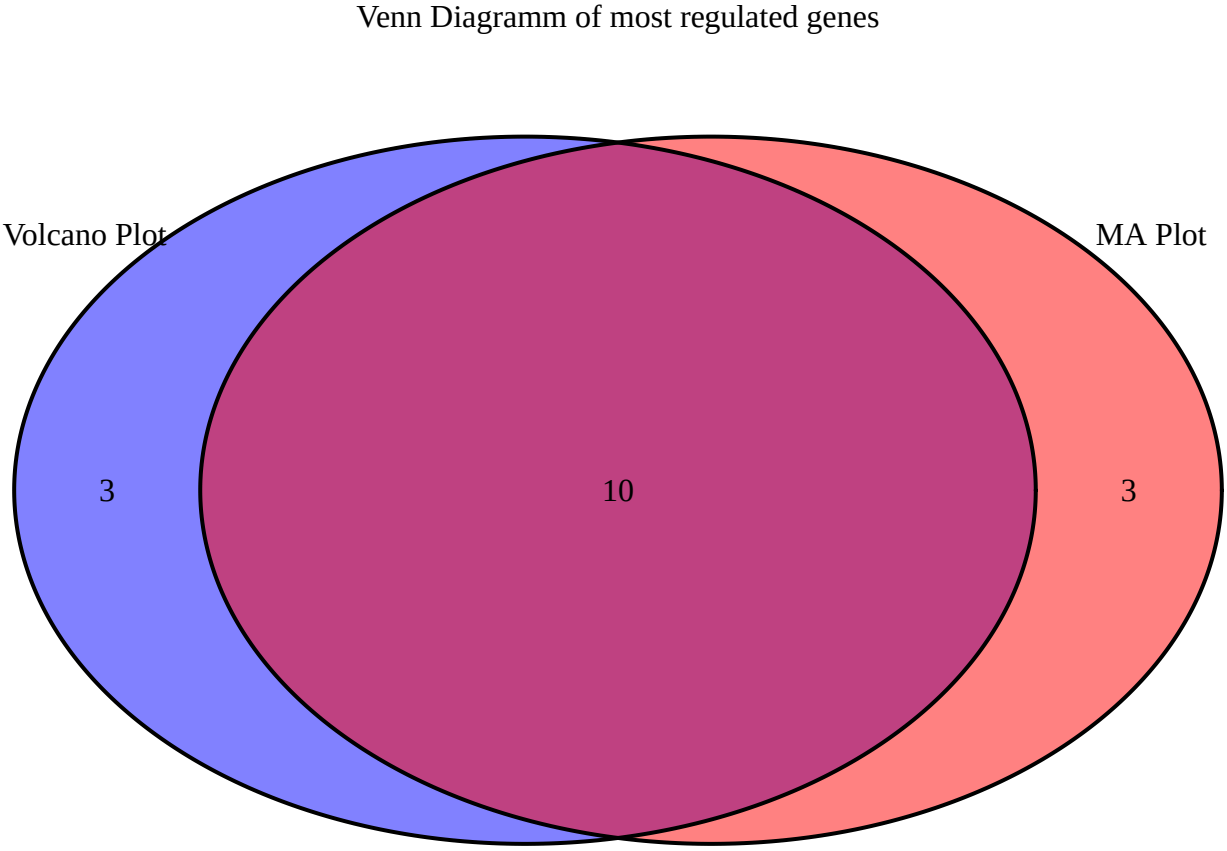
```
#MA plot labeled with important genes of MA plot
ggplot(data=MA)+
  aes(x=A, y=M, color= Significant)+
  geom_point()+
  xlab("mean expression")+
  ylab("log fold change")+
  ggtitle("MA plot of all genes")+
  geom_text(data=MA_labeled, aes(A, M, label=rownames(MA_labeled)))
```



Venn Diagram

Venn Diagramm is drawn to compare the most regulated genes by volcano plot and MA plot

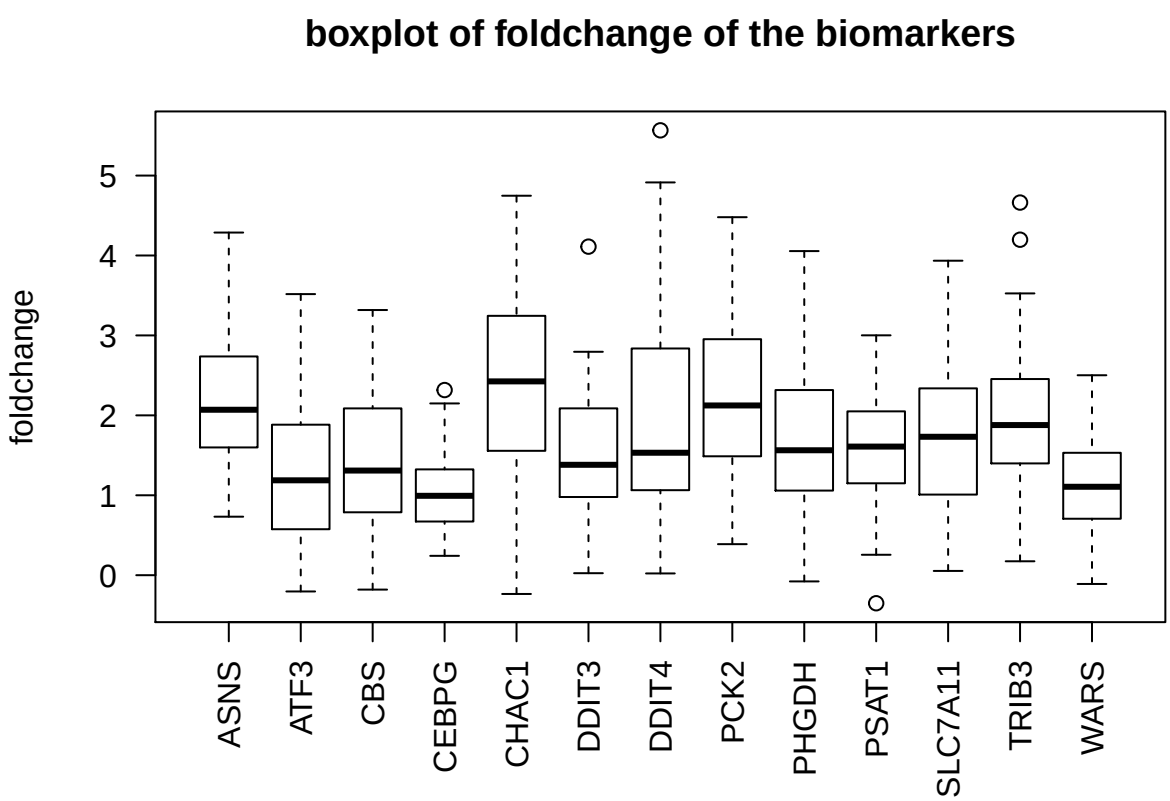
```
#Venn Diagram with biomarkers of volcano plot and MA plot
library(VennDiagram)
biomarkers_MA_vector <- rownames(MA_labeled)
venn.plot <- venn.diagram(
  x = list(
    "Volcano Plot" = biomarkers,
    "MA Plot" = biomarkers_MA_vector
    ),
  filename = NULL, fill = c("blue", "red"), main = "Venn Diagramm of most regulated genes"
  );
grid.draw(venn.plot);
```



Boxplot

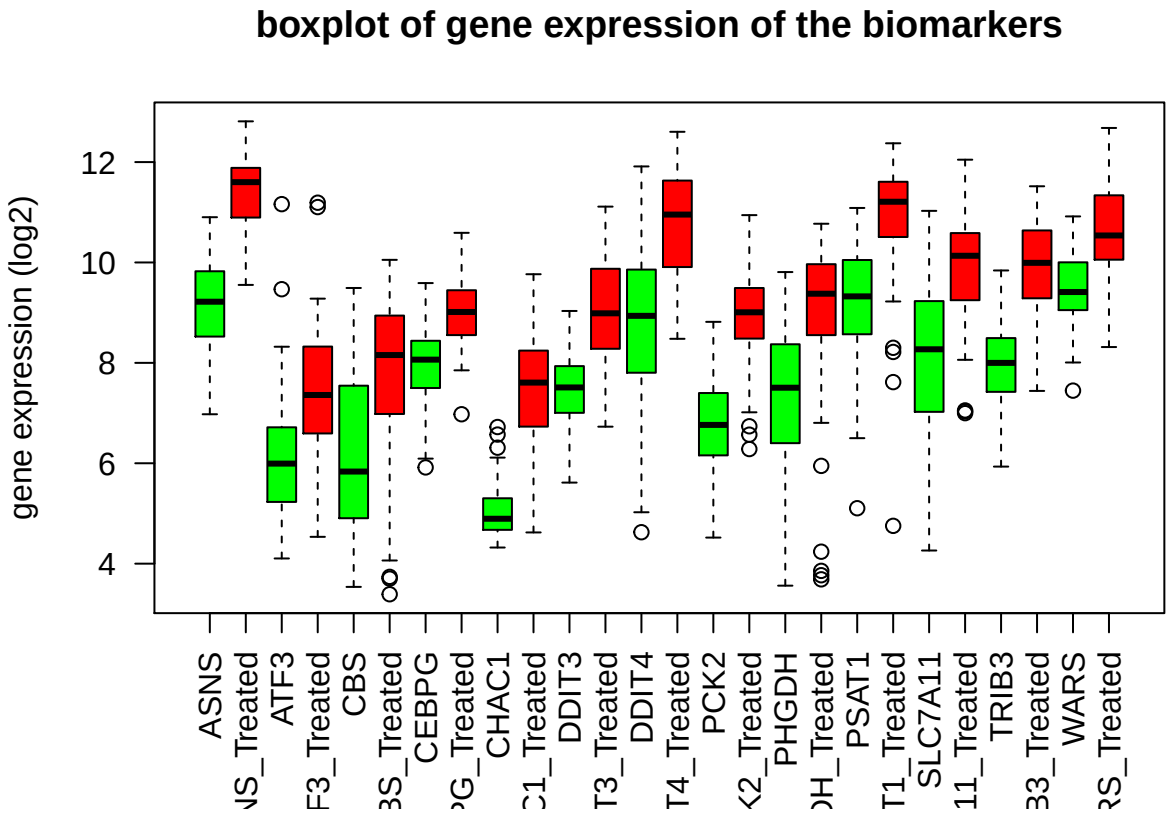
Draw a boxplot of the **foldchange** of biomarkers

```
# create a matrix foldchange_biomarkers, with the foldchange only of the biomarkers
foldchange_biomarkers <- sapply(biomarkers, function(x){
  e_foldchange[x, ]
})
boxplot(foldchange_biomarkers, ylab= "foldchange",
        main= "boxplot of foldchange of the biomarkers", las=2)
```



Draw a boxplot of the **untreated vs. treated gene expression** of biomarkers

```
# create a matrix e_treated_biomarkers/ e_untreated_biomarkers, with the gene expression only of the bi
e_treated_biomarkers <- sapply(biomarkers, function(x){
  e_treated[x, ]
})
e_untreated_biomarkers <- sapply(biomarkers, function(x){
  e_untreated[x, ]
})
colnames(e_treated_biomarkers) <- paste(colnames(e_treated_biomarkers),"Treated",
                                        sep = "_") #add treated to colnames

# create a matrix, which contains gene expression of untreated and treated and sort it after colnames
e_treated_untreated_biomarkers <- cbind (e_treated_biomarkers, e_untreated_biomarkers)
e_treated_untreated_biomarkers <- e_treated_untreated_biomarkers[,order(colnames(e_treated_untreated_bi

# create a color vector, where untreated samples are green and treated ones are red
color_boxplot_e_treated_untreated <- sapply(colnames(e_treated_untreated_biomarkers), function(x) {
  ifelse(x %in% grep ("Treated",colnames(e_treated_untreated_biomarkers), value = TRUE),
         "red", "green")})

# boxplot, where treated and untreated are right next to each other
boxplot(e_treated_untreated_biomarkers, ylab= "gene expression (log2)",
        main= "boxplot of gene expression of the biomarkers", las=2, col= color_boxplot_e_treated_untrea
```


boxplot of gene expression of the biomarkers
gene expression (log2)
12
10
8
6
4
ASNS
NS_Treated
ATF3
F3_Treated
CBS
3S_Treated
CEBPG
G_Treated
CHAC1
C1_Treated
DDIT3
T3_Treated
DDIT4
T4_Treated
PCK2
K2_Treated
PHGDH
DH_Treated
PSAT1
T1_Treated
SLC7A11
11_Treated
TRIB3
33_Treated
WARS
RS_Treated