# Data analysis: Project 2 Group 4

## Molecular Biotechnology, 4th term, Summer 2019

### Anna, Ann-Sophie und Jana

First of all, a broad analysis containing all samples of the NCI60 panel with 15 different cancer drugs was performed. In the specific analysis we focused on the drug erlotinib, which is an inhibitor of EGFR. Three milestones were defined to analyse which pathways are mostly regulated due to the drug treatment. The first one included finding the celllines which have the strongest fold change and the most regulated genes (biomarkers). Secondly, the correlation between the drug sensitivity of the various celllines (GI50) and the EGR1 expression was analysed. In the last part, we analysed the effect of erlotinib on different pathways using the package PROGENY and illustrating the fold change in gene expression due to the drug in a heatmap.

## Load data

Firstly, the data generated from the NCI60 cellline panel is downloaded and the various tables are saved as data frames.

```
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
The begin of the project includes some preparation of the data and annotation. Since the gene expression values are already logarithmic, the fold change caused by the drug can be calculated by substracting the untreated values (as a control of normal gene expression for each gene in each cellline) from the treated ones.

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

**Renaming of cellline SK-MEL-2**
One problem, which occured at the first structuring of the data by using the grep() function was that the cellline name SK-MEL-2 is part of cellline SK-MEL_28. To solve this we renamed that cellline to SK-MEL-2_.

```
#SK-MEL-2_ is added as new factor
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))
```

**Annotation** of all sample names
A matrix is created, which contains for each sample name the drug, cellline and cancertype. This matrix is later used for labeling and coloring of our plots.

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)
```

**Preparation for Coloring**
Create a vector which assigns each drug or each cancertype a color. These color vectors were used for coloring of the plots and creating the corresponding legends. For this purpose we searched for a color palette containing 15 colors, which are easy to distinguish. However, we only found some like RColorBrewer, which had no palette containing at least 15 distinguishable colors. Therefore we just defined our own color_palette by using the names of easily distinguishable colors.

1. Coloring 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
})
```

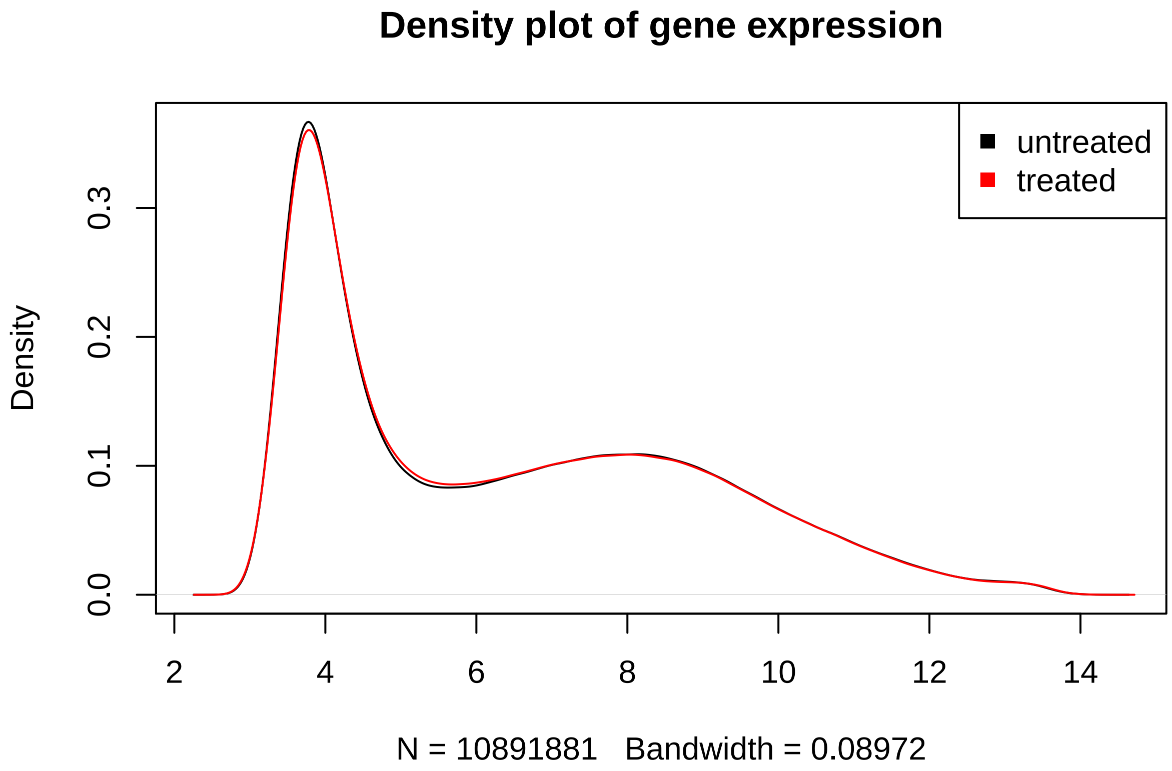
2. Coloring according to cancertype (color_vector_cancertype)

```
#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, 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)
```

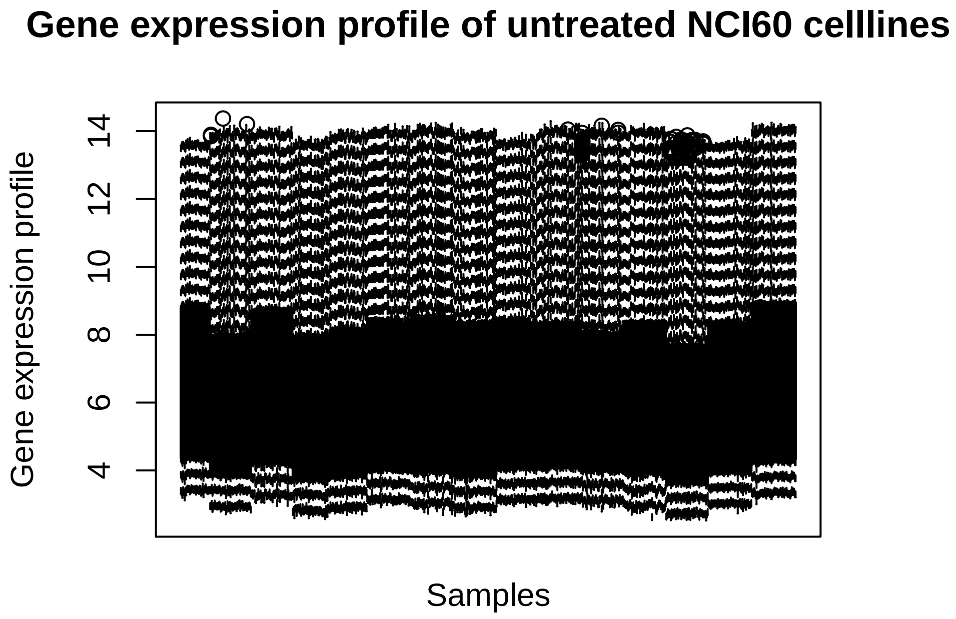


As expected, there can be hardly seen any difference between both curves. One reason for that is that the gene expression of most of the 13299 genes did not change due to the drug.

### Boxplot

In a next step the gene expression profile of each untreated sample was visualized in a boxplot to look, whether the complete expression profiles look the same over all samples or whether normalization is needed.

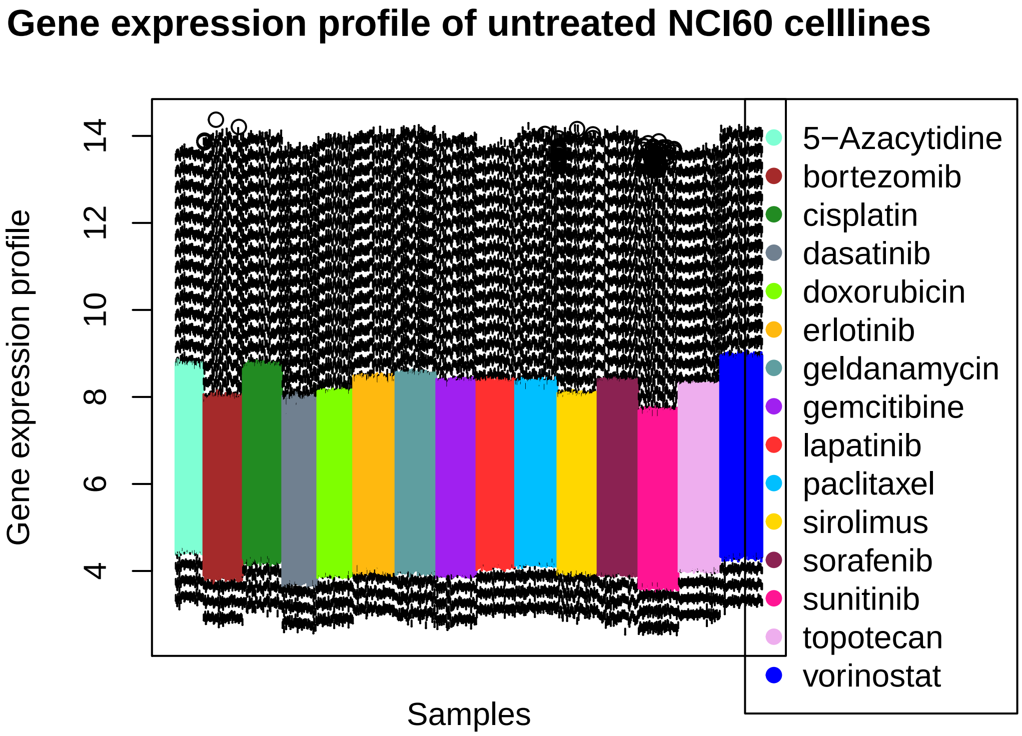
```
#par(oma = ) makes spaces outside the plot larger (oma = outer margins)
#xaxt = "n": removes labels on x-axis
#title() used to move xlab nearer to the axis
par(oma = c(1, 1, 1, 8), xpd = "TRUE")
boxplot(NCI_TPW_gep_untreated,
        xaxt = "n",
        ylab = "Gene expression profile",
        vertical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



In the boxplot sudden differences occuring with a regular pattern can be observed. An explanation for that could be that the gene expression of the samples was measured at different points of time or at different laboratories. This raised the question, whether these batches match with the 15 drugs these control measurements of untreated expressions were made. Therefor, the boxes were colored according to the drug the control was used for.

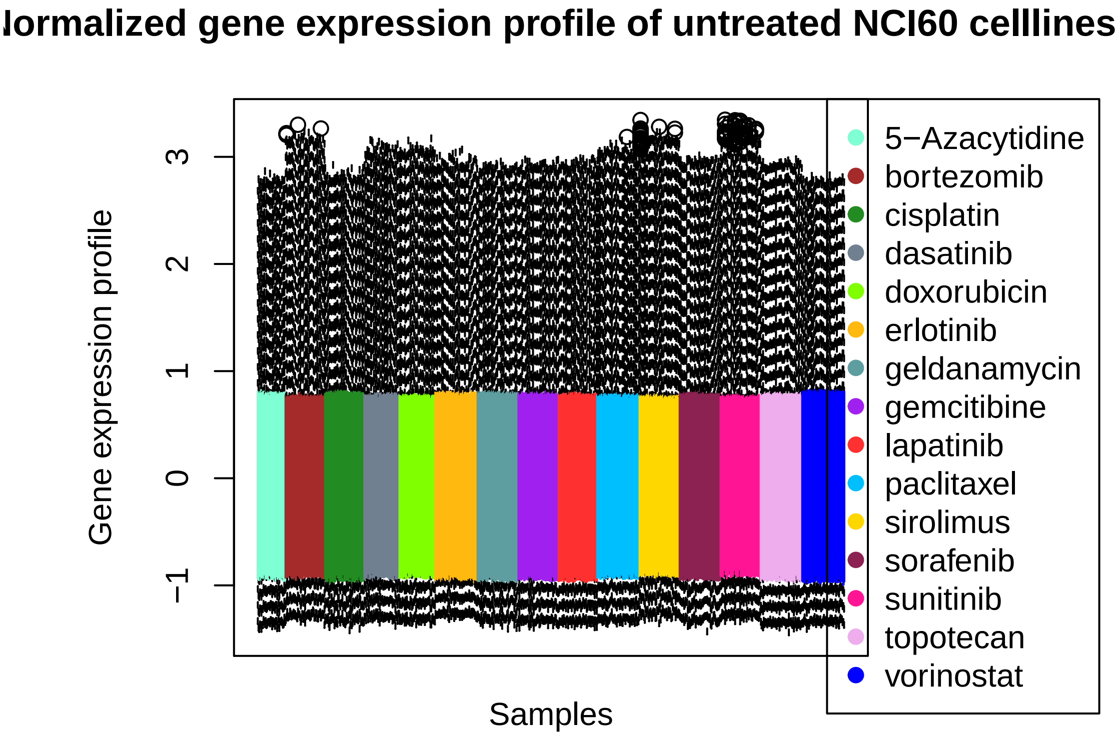
**Color plot according to drugs**

```
par(mar = c(4, 6, 4, 10), xpd = "TRUE")
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("topright",
       legend = names(color_palette_drug),
       col = color_palette_drug,
       inset = c(-0.365, 0),
       pch = 19)
```



Since the batches exactly match the different drugs for which the untreated expression was determined, a **normalization** is necessary if the values over the various drugs should be comparable.

```
#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(mar = c(4, 6, 4, 10), 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("topright",
       legend = names(color_palette_drug),
       col = color_palette_drug,
       pch = 19,
       inset = c(-0.365, 0))
```

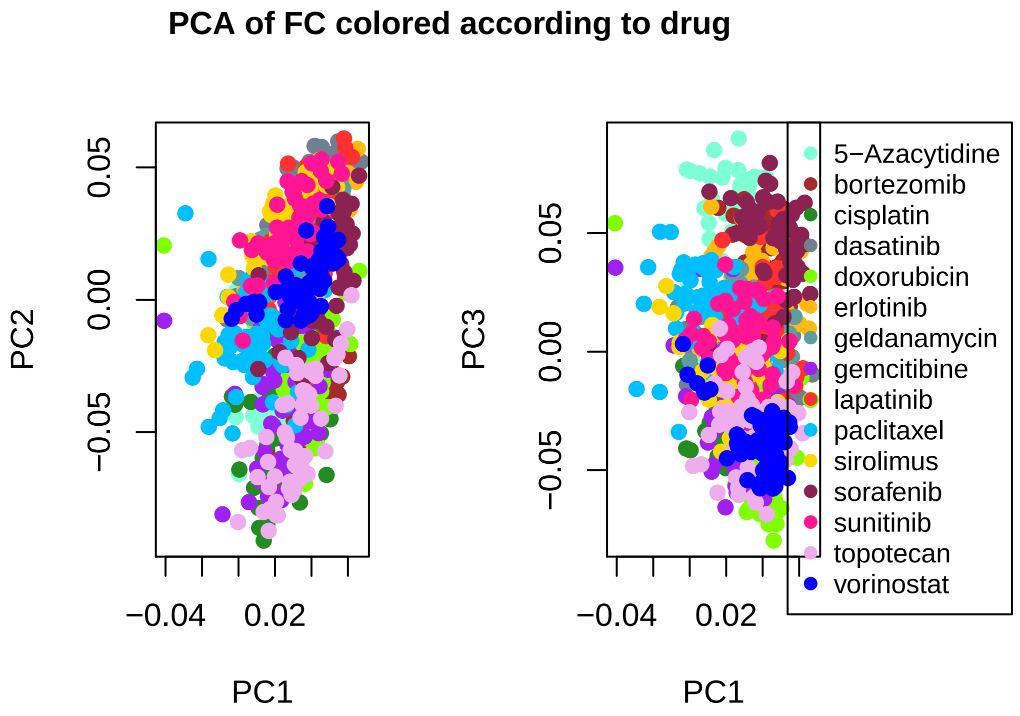


### PCA

A principal component analysis is used for dimensionality reduction. With these technique it is possible to depict most of the variance observed in the gene expression changes due to drug treatment (foldchange) over all samples. The points were colored firstly according to drug and secondly according to cancertype to see whether there are clusters corresponding to drug treatment or cancertype.

#### Coloring according to drug

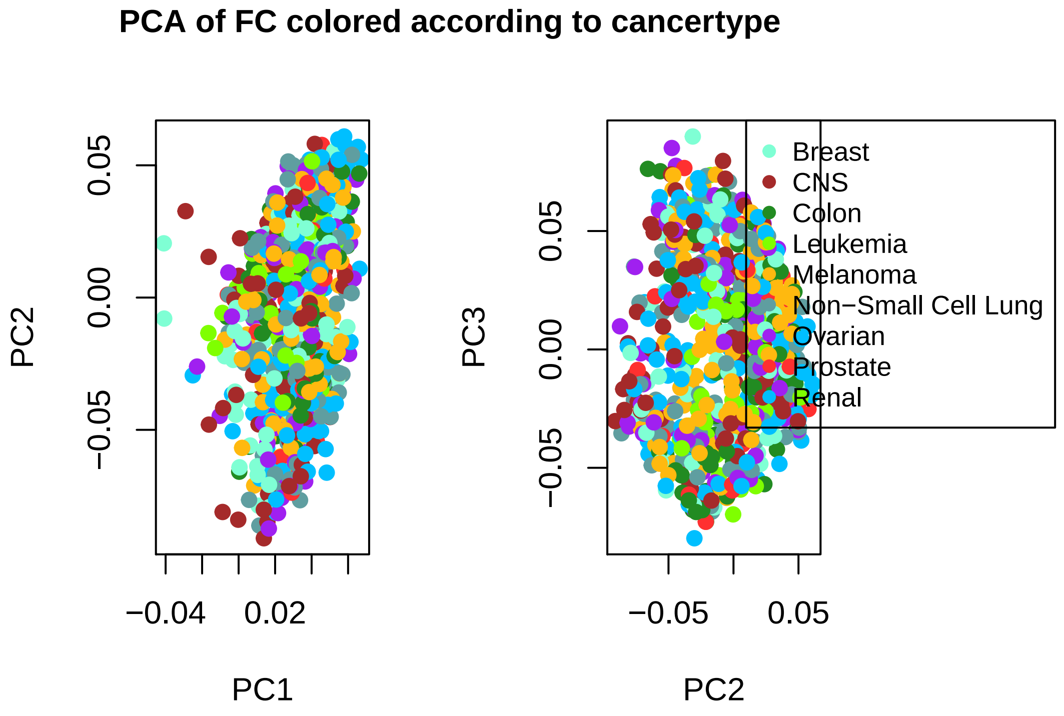
```
pca <- prcomp(FC_normalized)

par(oma = c(1, 1, 1, 8), mfrow = c(1, 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$rotation[,1],
     pca$rotation[,3],
     col = color_vector_drug,
     pch = 19,
     xlab = "PC1",
     ylab = "PC3")
#create legend on the right side
legend("topright",
       legend = names(color_palette_drug),
       col = color_palette_drug,
       pch = 19,
       xpd = "TRUE",
       cex = 0.8,
       inset = c(-0.9, 0))
#Title: mtext = margin text, side = 3 (upside)
mtext("PCA of FC colored according to drug",
      side = 3,
      line = -2,
      font = 2, #bold
      outer = TRUE)
```



In the PCA plot it can be seen that the celllines treated with the same drug accumulate in certain areas.

**Coloring according to cancertype**

```
par(oma = c(1, 1, 1, 8), mfrow = c(1, 2))
#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("topright",
       legend = names(color_palette_cancertype),
       col = color_palette_cancertype,
       pch = 19,
       xpd = "TRUE",
       inset = c(-1.1, 0),
       cex = 0.8)
mtext("PCA of FC colored according to cancertype",
      side = 3,
      line = -2,
      font = 2, #bold
      outer = TRUE)
```



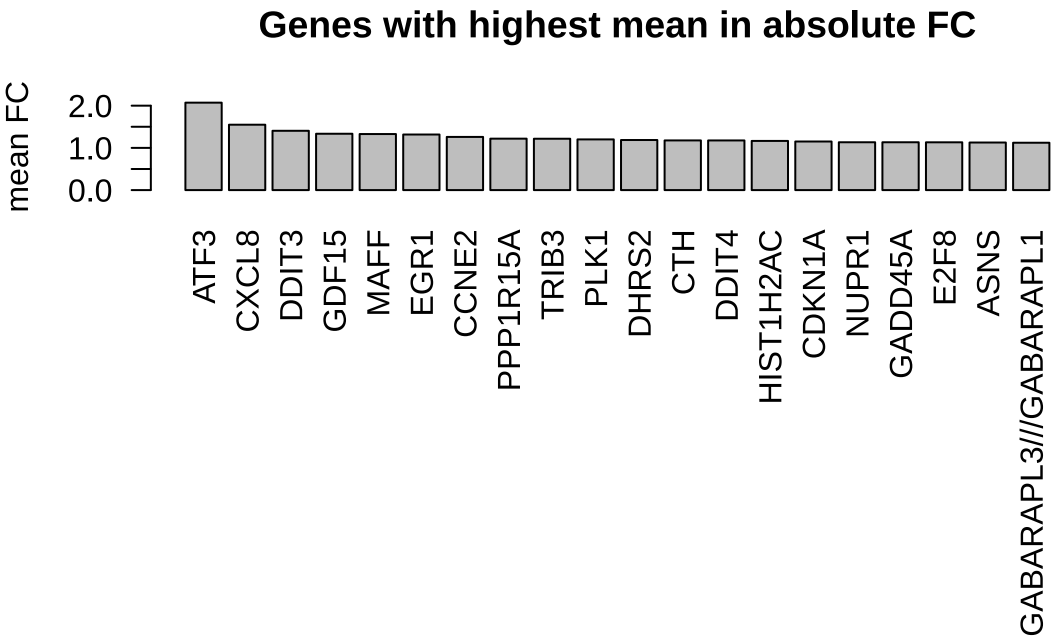
```
rm(pca)
```

The colors showing which cancertype a cellline belongs to seem rather random distributed in the PCA plot. No clustering between the celllines beeing part of the same cancertype can be observed.

### Most regulated genes

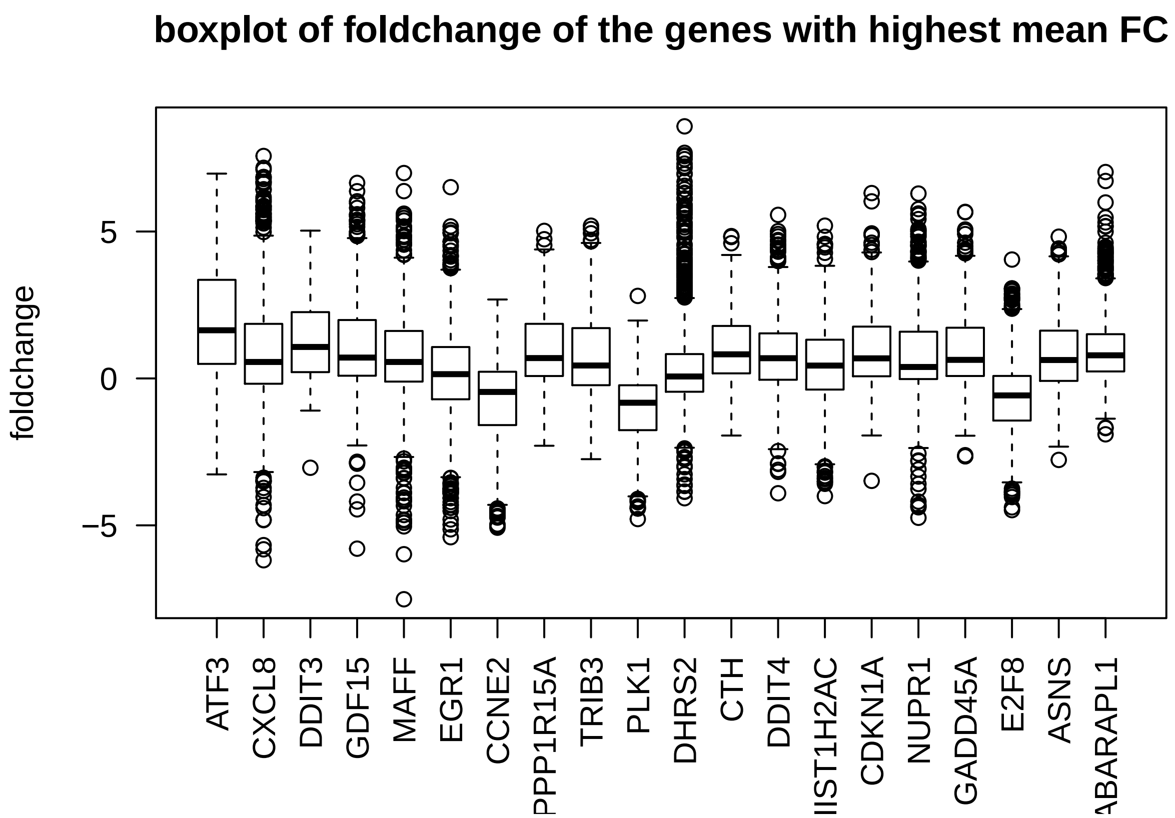
**Barplot** to find genes, which were mostly regulated by all cancer treatments

```r
#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

```r
#FC_samples_with_highest_mean_FC contains the gene expression of the 20 biomarkers (20 columns) of all
FC_samples_with_highest_mean_FC <- data.matrix(as.data.frame(sapply(names(mean_FC_abs)[1:20], function(
  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)
```



## 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

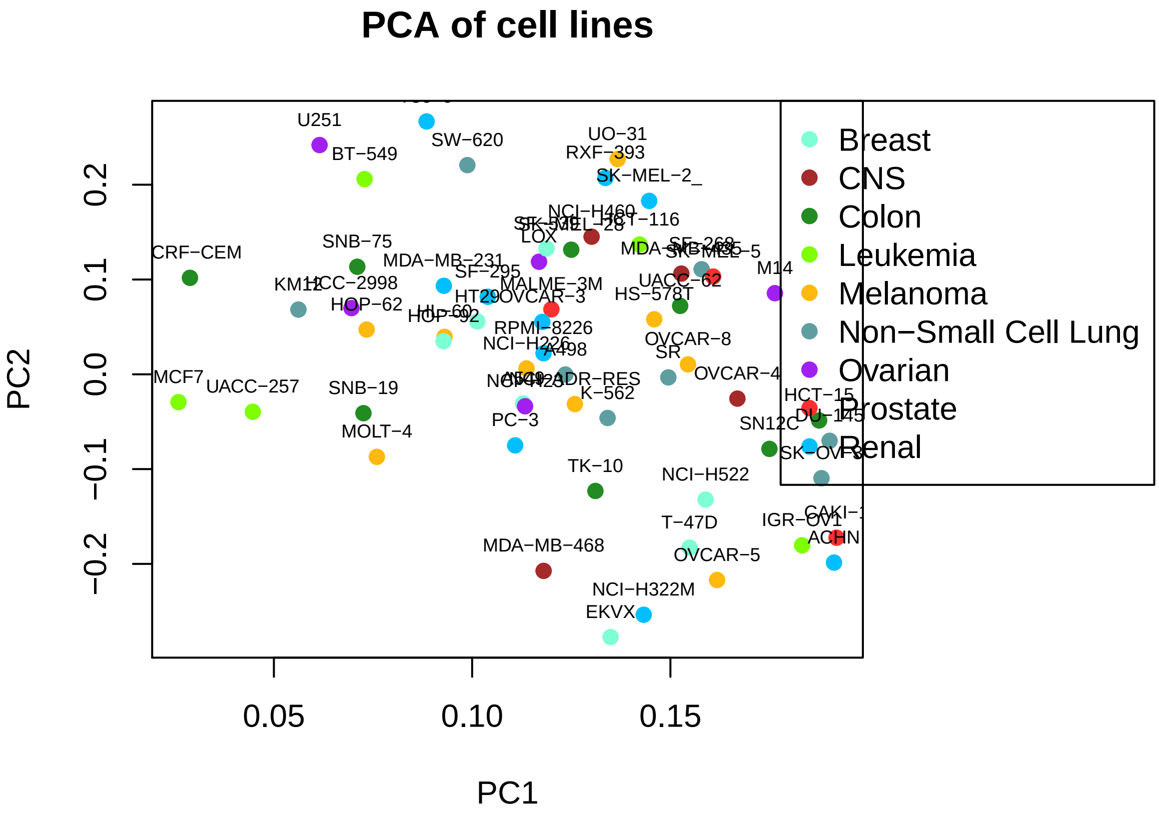
```
#select 15 cell lines with highest variance (greater than 75% quantile, sorted by decreasing value)
var_cell_line <- apply(e_foldchange, 2, var)
cell_line_var_greater_75quantile <- sort(var_cell_line [which (abs(var_cell_line) > quantile(abs(var_cel
cell_line_var_greater_75quantile <- round(cell_line_var_greater_75quantile, 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[,2],
     col=color_vector_cancertype,
     pch=19,
     xlab = "PC1",
     ylab="PC2",
     main = "PCA of cell lines")
legend("topright",
       legend = names(color_palette_cancertype),
       col = color_palette_cancertype,
       pch = 19,
       xpd = TRUE,
       inset = c(-0.41, 0))
#label points
text(pca$rotation[ ,1],
     pca$rotation[ ,2],
     colnames(e_foldchange_normalized),
     pos = 3,
     cex = 0.6)
```

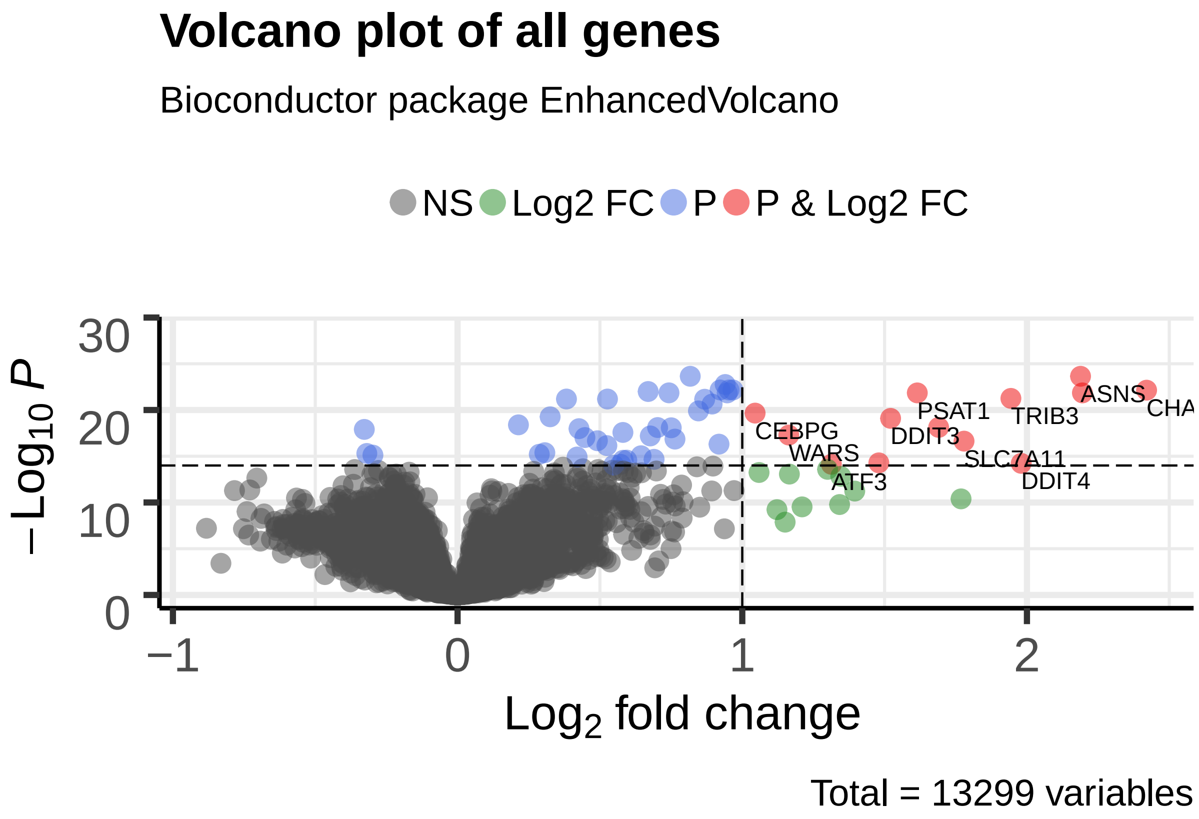


### Most regulated genes

#### Volcano plot

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

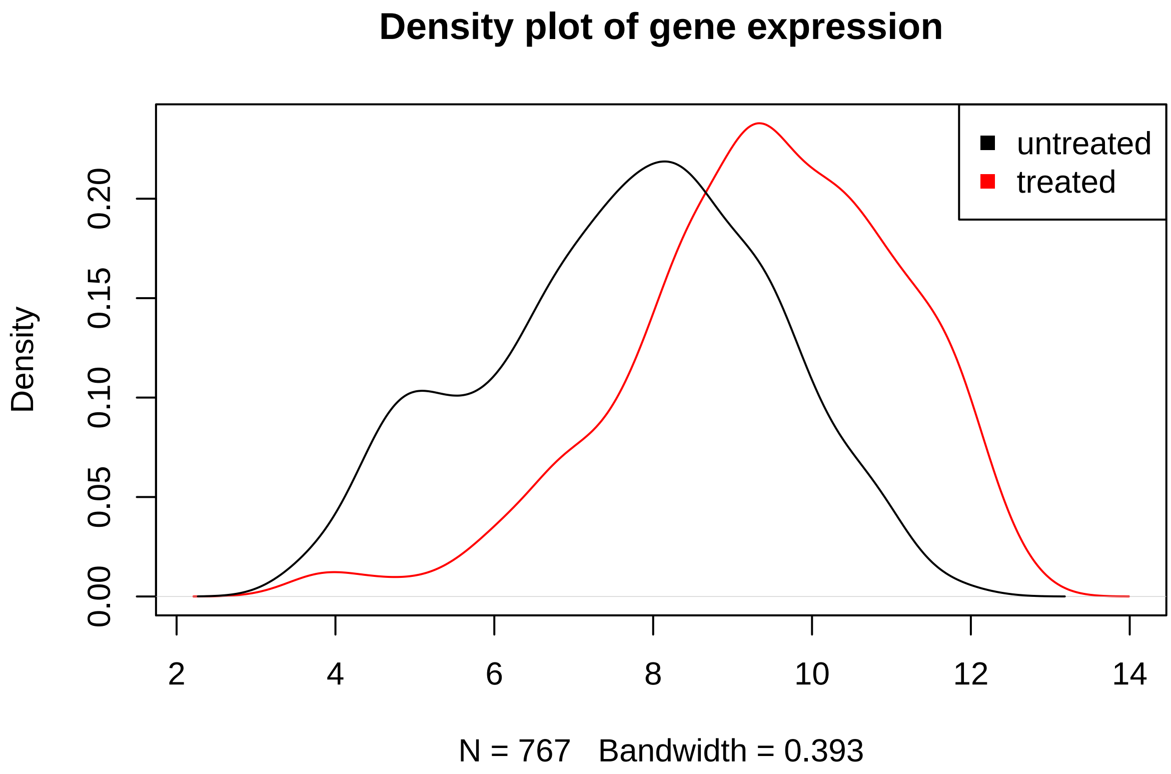
```r
#mean of gene expression of each gene over all cell lines
e_foldchange_mean_over_cell_lines <- rowMeans(e_foldchange) #equal to e_treated_mean_over_cell_lines -

#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 = 3.0)
```



### 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)]

#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)
```

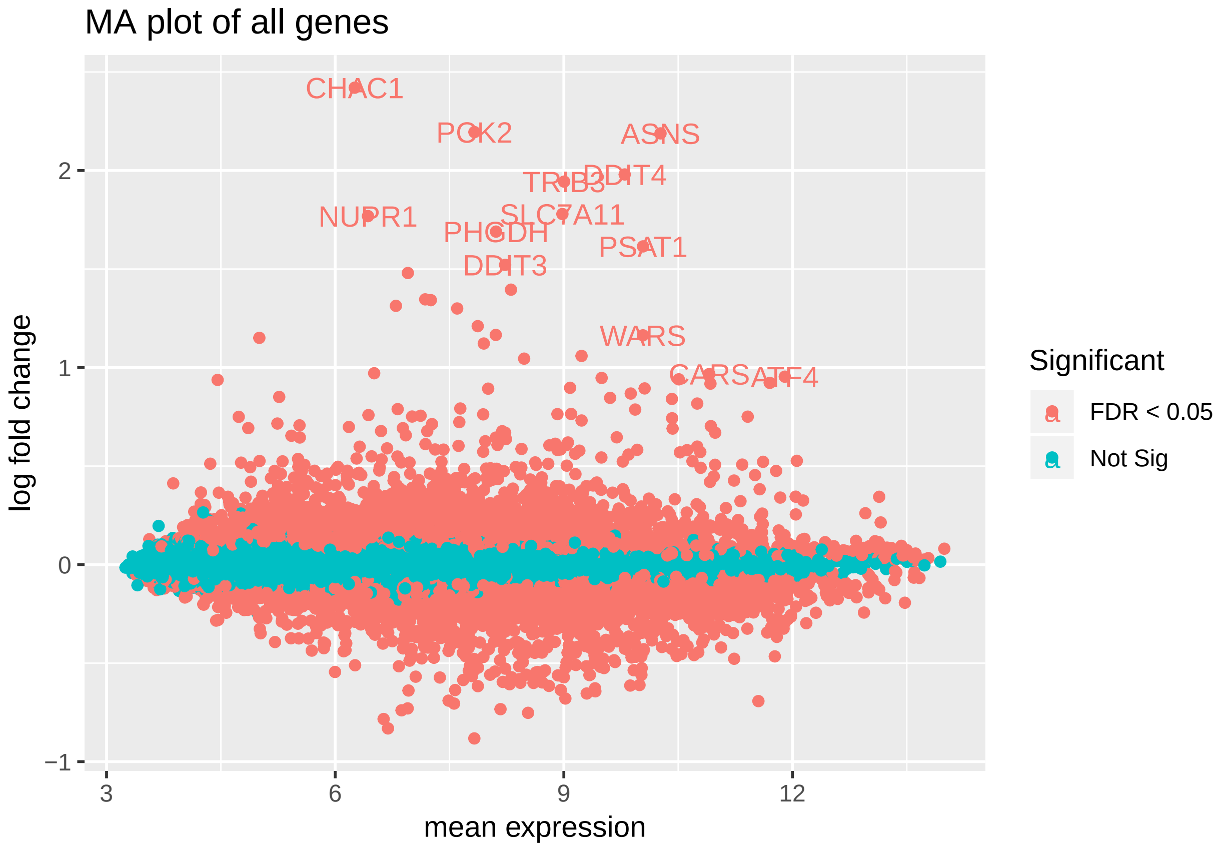


### MA-Plot

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

```
#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)
```

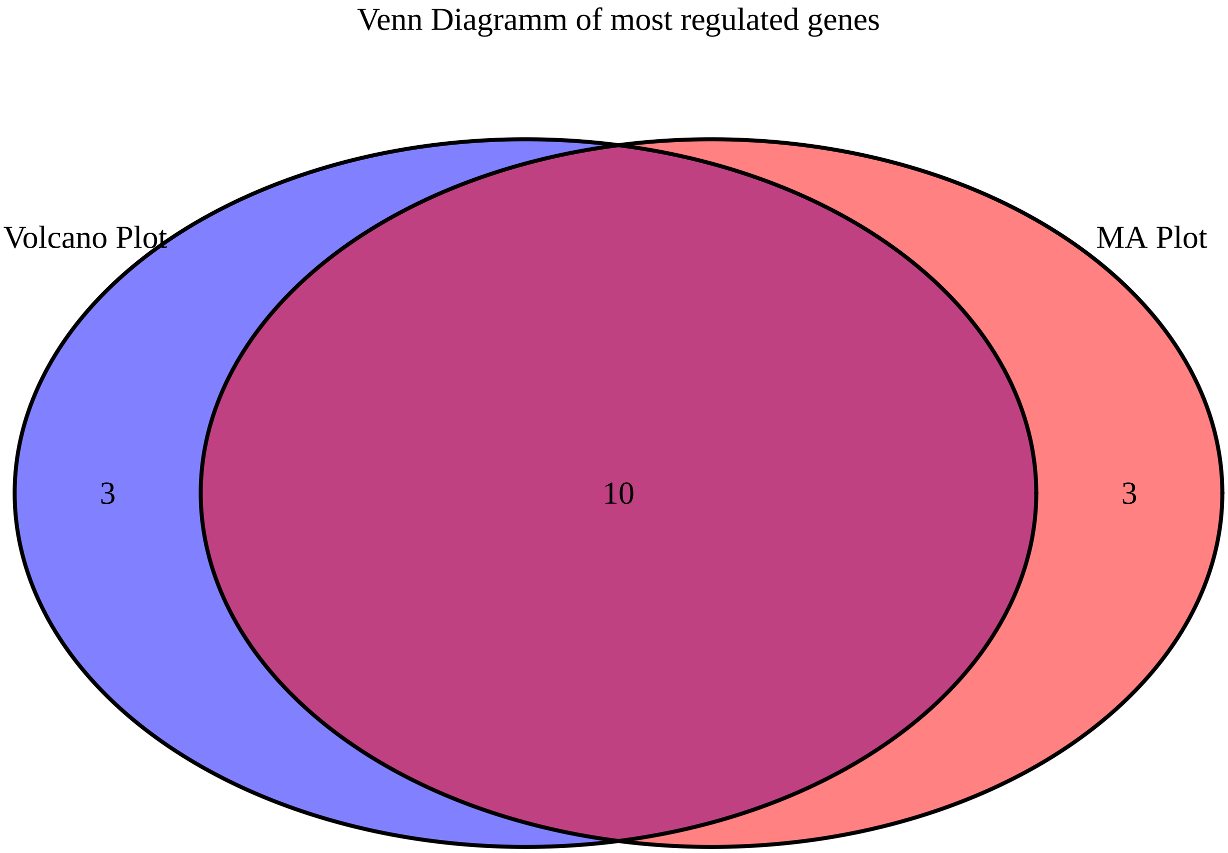
```
rm(M, A)
MA <- as.data.frame(MA)
MA$Significant <- ifelse(MA$FDR_values<0.05, "FDR < 0.05", "Not Sig")

#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

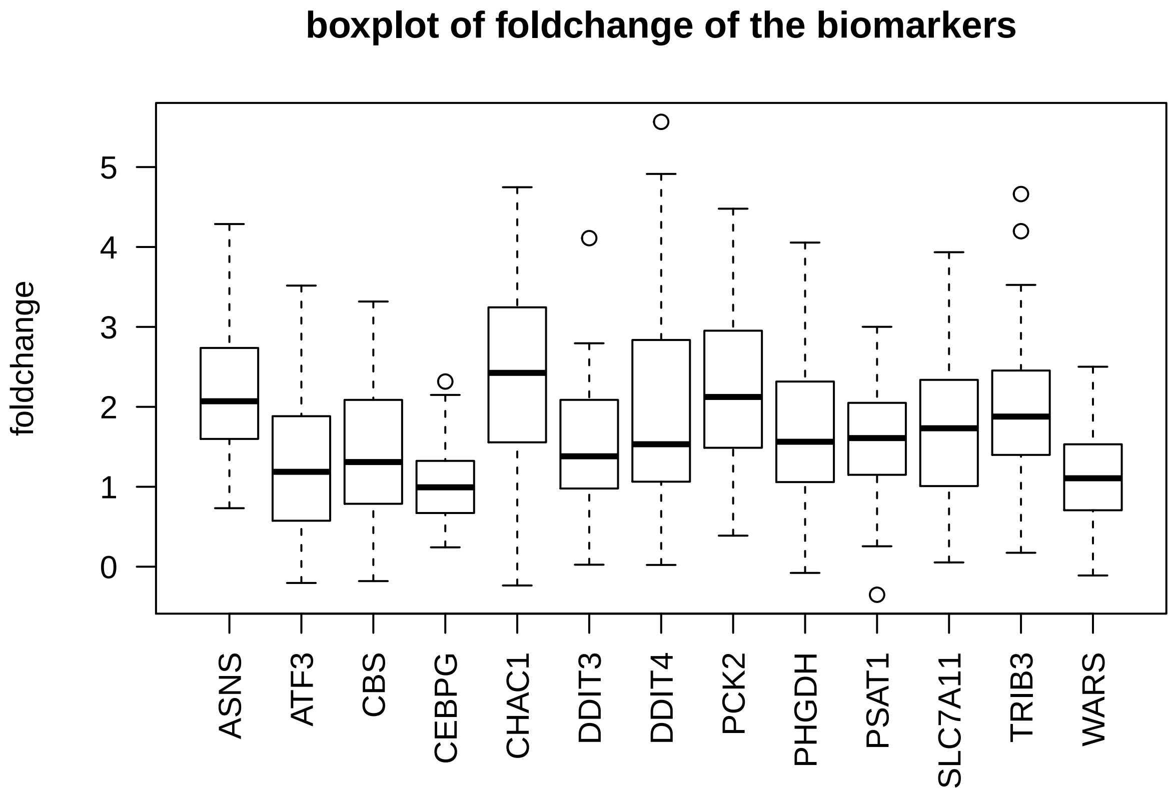
```
#Venn Diagram with biomarkers of volcano plot and MA plot
library(VennDiagram)
biomarkers_MA_vector <- rownames(MA_labeled)
venn.plot <- venn.diagram(
  x = list(
    "Volcano Plot" = biomarkers,
    "MA Plot" = biomarkers_MA_vector
    ),
  filename = NULL, fill = c("blue", "red"), main = "Venn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



### Boxplot

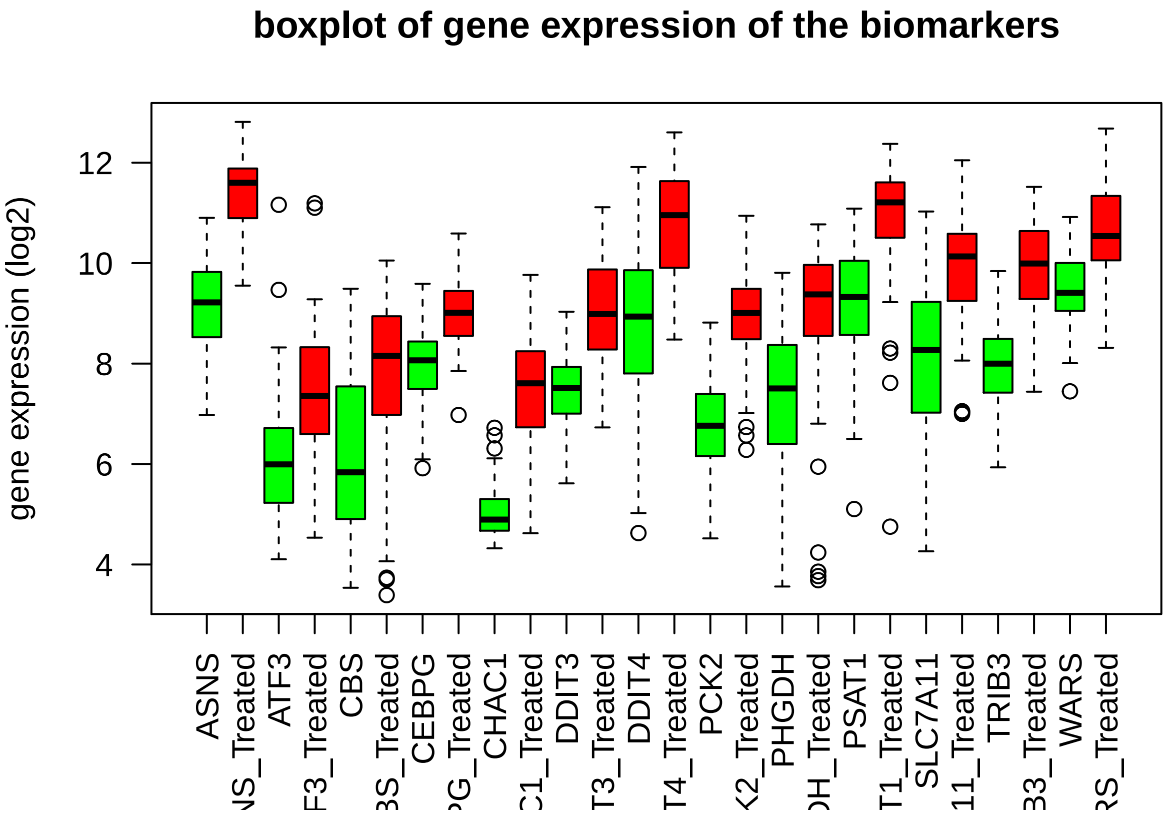
Draw a boxplot of the **foldchange** of biomarkers

```
# 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

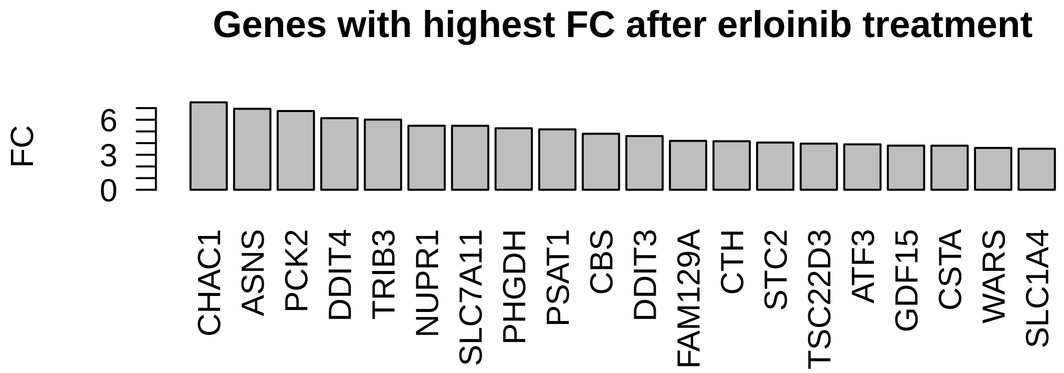
```
# create a matrix e_treated_biomarkers/ e_untreated_biomarkers, with the gene expression only of the bi
e_treated_biomarkers <- sapply(biomarkers, function(x){
  e_treated[x, ]
})
e_untreated_biomarkers <- sapply(biomarkers, function(x){
  e_untreated[x, ]
})
colnames(e_treated_biomarkers) <- paste(colnames(e_treated_biomarkers),"Treated",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



### 3. Milestone - Does the fold change of specific genes correlate with cell growth inhibition?

Draw a barplot of the foldchange of genes which have the highest foldchange after erlotinib treatment

```
#Barplot of genes with highest mean in FC over Erlotinib
genes_FC_erlotinb <- apply(e_foldchange_normalized, 1, mean)
genes_FC_erlotinb <- sort(abs(genes_FC_erlotinb), decreasing = TRUE)
par(oma =c(10,1,1,1))
barplot(genes_FC_erlotinb [1:20], main = "Genes with highest FC after erloinib treatment", ylab = "FC",
```



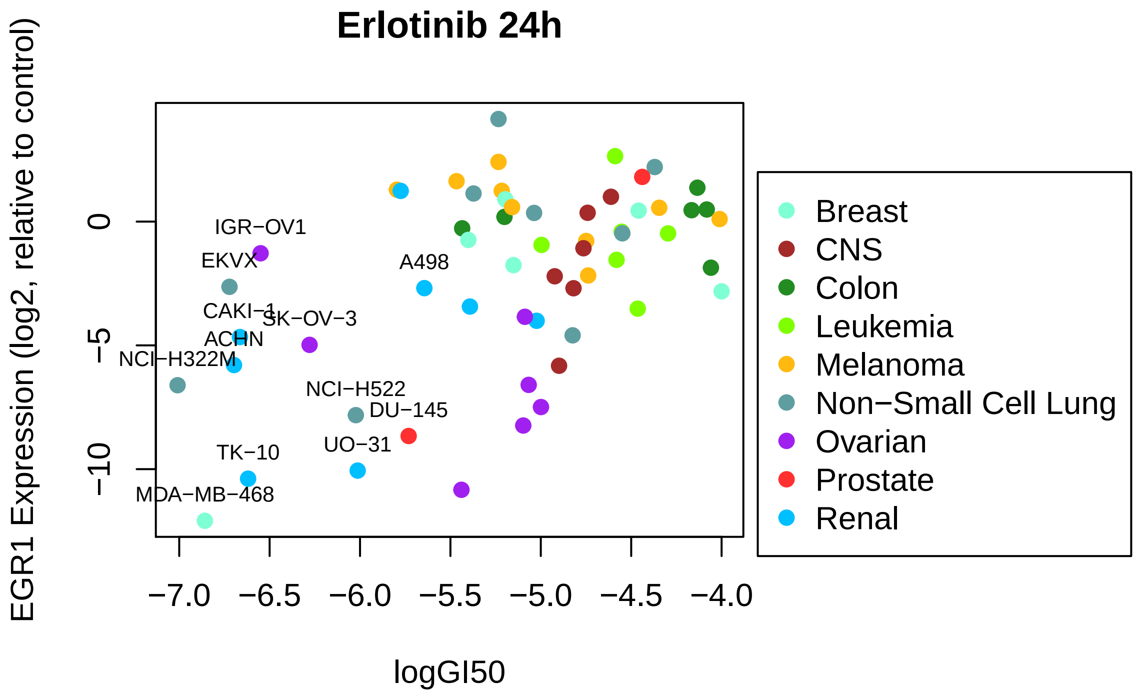
#### Data prepatation

```
# vector which only includes celllines which were used in e_foldchange_normalized
NegLogGI50_59_celllines <- NegLogGI50 ["erlotinib", -c(8,29)]
#@Anna: should we change the name from NegLogGI50_59_celllines_neg to simply LogGI50?
NegLogGI50_59_celllines_neg <- NegLogGI50_59_celllines * (-1)
```

#### Scatter plot: Relation between GI50 and EGR1 expression

Draw a scatterplot which include the GI50 values against the EGR1 expression revative to the untreated control. EGR1 is a transcriptional factor and is assocciated with the activiation of tumor suppressor genes like p53/TP53 and TGFB1, and plays an important role in the regulation of growth factor responses.

```
#Coloring according to cancertype
e_color_cancertype <- color_vector_cancertype[grep("erlotinib", names(color_vector_cancertype), value =

#Scatter plot
par(oma = c(1,1,1,10), xpd = "TRUE") #size of outer margins: bottom, top, left, right
plot(NegLogGI50_59_celllines_neg, e_foldchange_normalized ["EGR1",],
     col = e_color_cancertype,
     pch = 19,
     xlab = "logGI50",
     ylab = "EGR1 Expression (log2, relative to control)",
     main = "Erlotinib 24h")

legend(x = -3.8, y = 2,
       legend = names(color_palette_cancertype),
       col = color_palette_cancertype,
       pch = 19)

#label only points in the left bottom quarter
labeled_celllines <- names(NegLogGI50_59_celllines_neg)[NegLogGI50_59_celllines_neg < - 5.5
                                                         & e_foldchange_normalized["EGR1", ] < 0]

text(NegLogGI50_59_celllines_neg[labeled_celllines], e_foldchange_normalized ["EGR1", labeled_celllines]
     labels = labeled_celllines,
     cex = 0.7,
     pos = 3) #position of text at the top of the point
```



###Pearson correlation

```
#Pearson correlation
res <- cor.test(NegLogGI50_59_celllines_neg, e_foldchange_normalized ["EGR1",],
                method = "pearson")
res
```

```
##
##  Pearson's product-moment correlation
```

```
##
## data:  NegLogGI50_59_celllines_neg and e_foldchange_normalized["EGR1", ]
## t = 4.4344, df = 57, p-value = 4.265e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2876774 0.6749907
## sample estimates:
##       cor
## 0.5064499
```

###linear regression between EGR1 expression and the GI50 value after erlotinib treatment

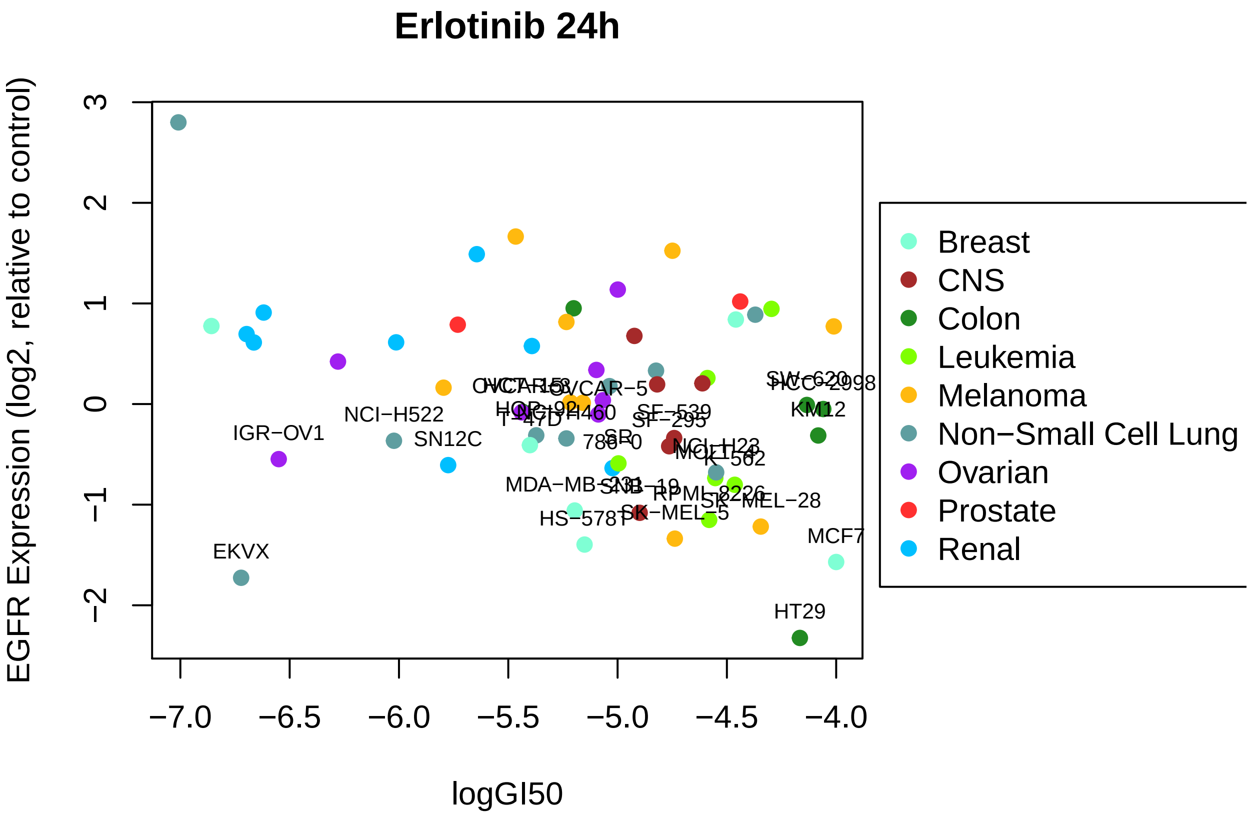
```r
linearMod_EGR1 <- lm(NegLogGI50_59_celllines_neg ~ e_foldchange_normalized ["EGR1",])  # build linear r

summary(linearMod_EGR1)
```

```
##
## Call:
## lm(formula = NegLogGI50_59_celllines_neg ~ e_foldchange_normalized["EGR1",
##     ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5112 -0.4494  0.1572  0.5239  1.2300
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       -4.93538    0.10250 -48.151  < 2e-16 ***
## e_foldchange_normalized["EGR1", ]  0.10453    0.02357   4.434 4.27e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6792 on 57 degrees of freedom
## Multiple R-squared:  0.2565, Adjusted R-squared:  0.2434
## F-statistic: 19.66 on 1 and 57 DF,  p-value: 4.265e-05
```

Only 25 % can be described by EGR1- expression, which makes a linear regression model kind of unfitted.

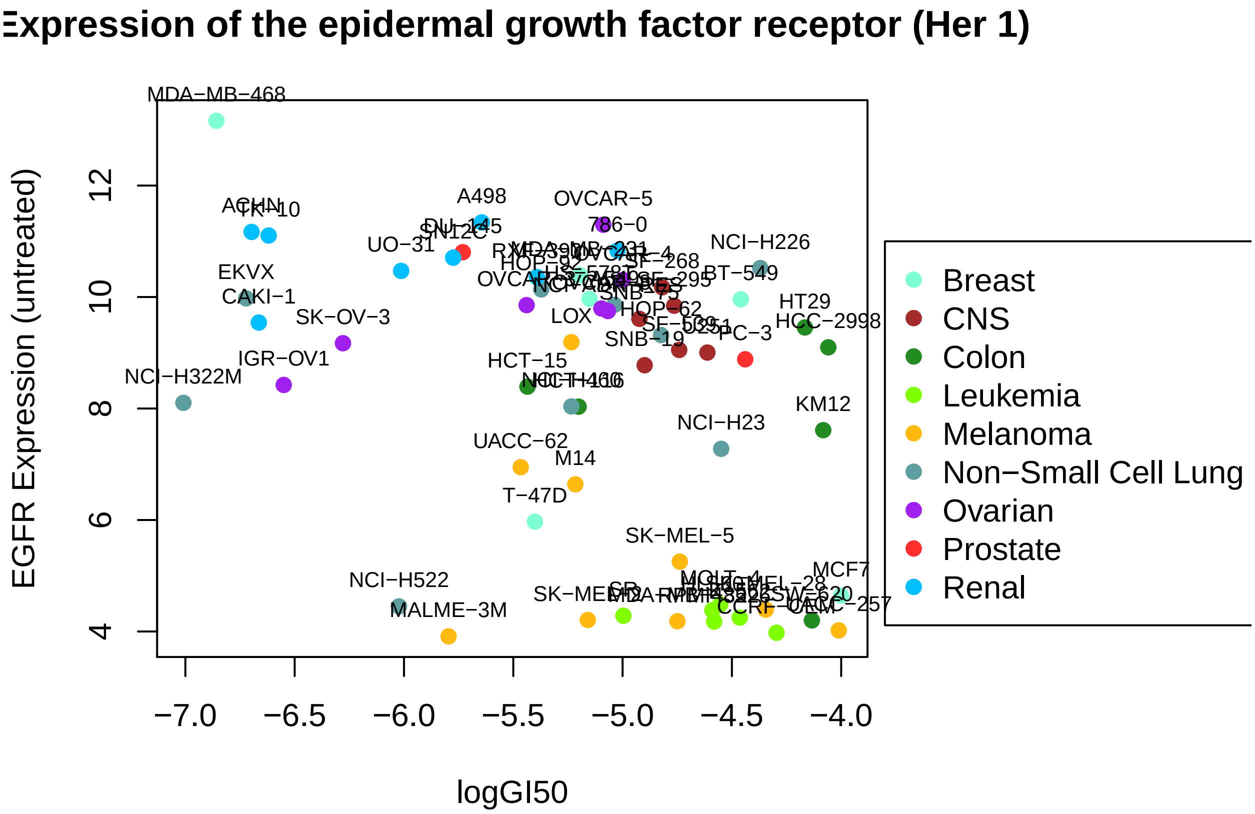
**Expressionsdaten von EGFR Expression relativ to control –> Does erlotinib treatment affect her1/ EGFR expression?**

```r
par(mar = c(4,4,4,10), xpd = "TRUE")
plot(NegLogGI50_59_celllines_neg, e_foldchange_normalized ["EGFR",],
      col = e_color_cancertype,
      pch = 19,
      xlab = "logGI50",
      ylab = "EGFR Expression (log2, relative to control)",
      main = "Erlotinib 24h")

legend(x = -3.8, y = 2,
      legend = names(color_palette_cancertype),
      col = color_palette_cancertype,
      pch = 19)

#label only points which have a decreased Her 1 expression after erlotinib treatment
labeled_celllines <- names(NegLogGI50_59_celllines_neg)[ e_foldchange_normalized["EGFR", ] < 0]

text(NegLogGI50_59_celllines_neg[labeled_celllines], e_foldchange_normalized ["EGFR", labeled_celllines]
      labels = labeled_celllines,
      cex = 0.7,
      pos = 3) #position of text at the top of the point
```



**Her 1 gene expression in the untreated celllines against GI50 values**

```
par(mar = c(4,4,4,10), xpd = "TRUE")
plot(NegLogGI50_59_celllines_neg, e_untreated ["EGFR",],
      col = e_color_cancertype,
      pch = 19,
     xlab = "logGI50",
     ylab = "EGFR Expression (untreated)",
     main = "Expression of the epidermal growth factor receptor (Her 1)")
legend(x = -3.8, y = 11,
    legend = names(color_palette_cancertype),
    col = color_palette_cancertype,
    pch = 19,
    xpd = TRUE)

text(NegLogGI50_59_celllines_neg, e_untreated ["EGFR",],
     labels  = names(NegLogGI50_59_celllines_neg),
     cex = 0.7,
     pos = 3)

#Funktioniert noch nicht
abline(lm(NegLogGI50_59_celllines_neg ~ e_untreated["EGFR",]))
```



```
#ERROR
# lable celline which belong to cluster 3 in the heatmap in milestone 4
#labeled_cellline_clusters <- names(NegLogGI50_59_celllines_neg)[ cellline_clusters [,1]  > 2 ]
#text(NegLogGI50_59_celllines_neg[labeled_cellline_clusters], e_untreated[ "EGFR",],
    #labels = labeled_cellline_clusters,
    #cex = 0.7,
    #pos = 3)
```

### lineare regression Her 1 expression against the GI50 values

```
linearMod_Her1 <- lm(NegLogGI50_59_celllines_neg ~ e_untreated ["EGFR",])  # build linear regression mo
summary(linearMod_Her1)
```

```
##
## Call:
## lm(formula = NegLogGI50_59_celllines_neg ~ e_untreated["EGFR",
##     ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.85461 -0.30720  0.09961  0.47793  1.22004
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -4.14018    0.31366 -13.199  < 2e-16 ***
## e_untreated["EGFR", ] -0.12518    0.03657  -3.423  0.00115 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7174 on 57 degrees of freedom
## Multiple R-squared:  0.1705, Adjusted R-squared:  0.156
## F-statistic: 11.72 on 1 and 57 DF,  p-value: 0.001152
```

mode does nott fit that well, only 17% can be disrecped by Her1 expression. Maybe we have to consider all types of Her receptors 1-4!!! cause the composition of Her1 and Her3/4 play a part when it comes to bad/good prognoses an therfore could play a role in the GI50 values an the success of erlotinib treatment.