DE analysis - Day1

Sergey Naumenko

2021-05-08

Contents

Overview

- Principal Investigator: Beth Overmoyer
- Experiment: RNAseq_analysis_of_inflammatory_breast_cancer_hbc04141
- study 6 was excluded because if low read depth in 3373-3
- https://www.bioconductor.org/packages/release/bioc/vignettes/DEGreport/inst/doc/DEGreport.html
- AnnotationHub. We use ensembl version matching bcbio pipeline - v94.
- HBC materials
- HBC materials - functional analysis
- http://bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html
- this is DE for Day1 samples

Checking to see that the transcript to gene mapping is correct

When you have annotations that are from a different source from your reference you can run into problems (i.e lose genes). Some checks you can do before proceeding:

1. Look at the dimensions of your count matrix. Do you have ~20k genes present? `dim(txi$counts)`
2. When running `tximport()` you will get a message in your console. If you see something like `transcripts missing from tx2gene` start troubleshooting.

```
dim(txi$counts)
```

```
## [1] 58735    22
```

Sanity check that metadata matches your expression

It is always a good idea to check if:

1. Do you have expression data for all samples listed in your metadata?
2. Are the samples in your expression data in the same order as your metadata?

```
### Check that sample names match in both files
all(colnames(txi$counts) %in% rownames(meta))
```

```
## [1] TRUE
```

```
### Check that sample names match in both files
all(colnames(txi$counts) %in% rownames(meta))
```

```
## [1] TRUE
```

```
### Check that all samples are in the same order
meta <- meta[colnames(txi$counts),]
all(colnames(txi$counts) == rownames(meta))
```

```
## [1] TRUE
```

Run DESeq2

```
estimating size factors
estimating dispersions
gene-wise dispersion estimates
mean-dispersion relationship
final dispersion estimates
fitting model and testing
```

- Estimating size factors and count normalization
- Gene-wise dispersions
- Mean-dispersion(variance) relationship and the Negative Binomial Model
- Model fitting and hypothesis testing

Wald test

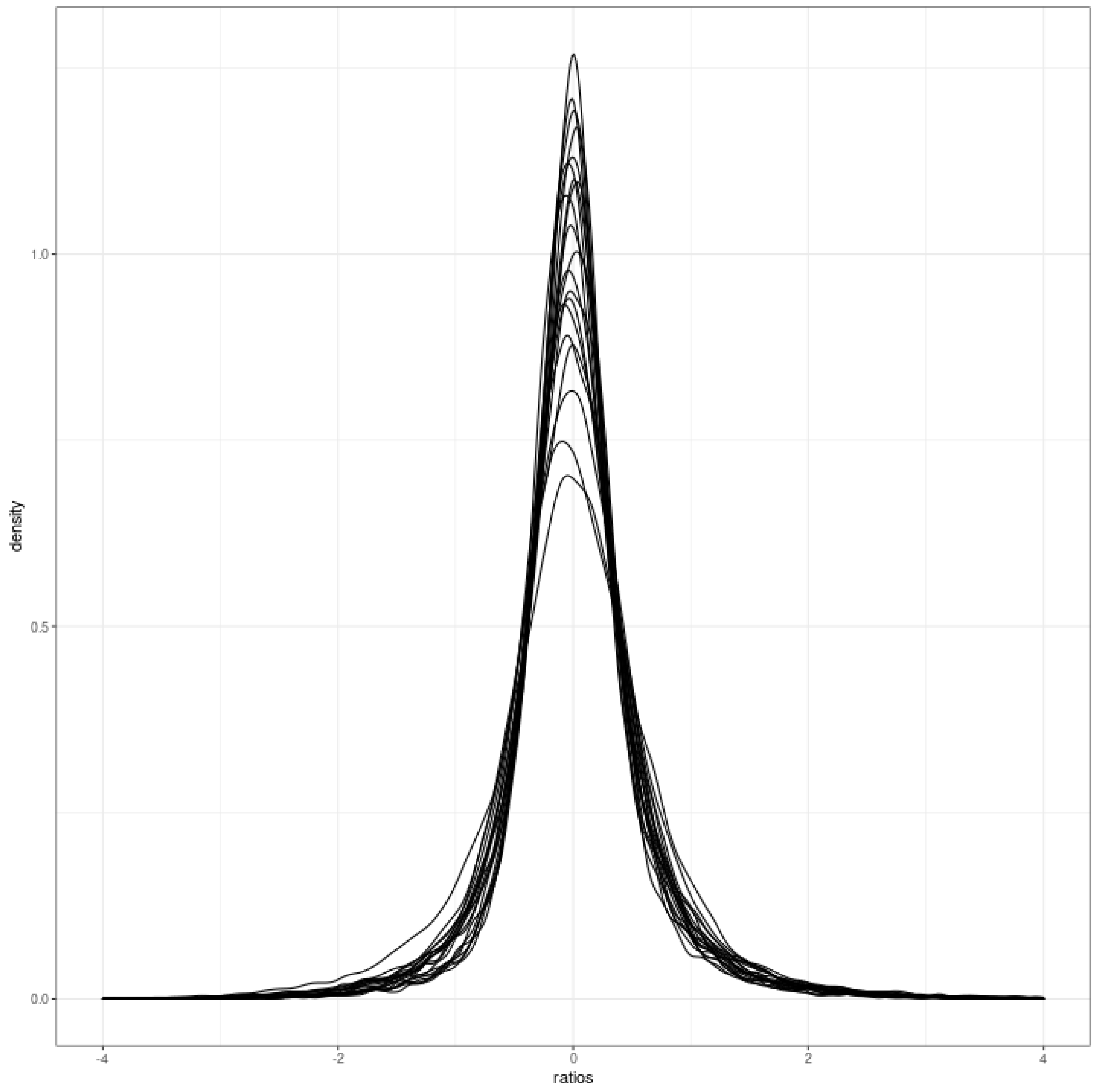
Here we subset protein coding genes.

```
## Create DESeq2Dataset object
dds_file <- "data/dds.day1.RDS"
meta$treatment <- as.factor(meta$treatment)
meta$response <- as.factor(meta$response)
meta$er <- as.factor(meta$er)
meta$date_of <- as.factor(meta$date_of)
meta$tumor_percentage <- as.factor(meta$tumor_percentage)
meta$tumor_percentage_high <- as.factor(meta$tumor_percentage_high)

if (remove_cases_2_19){
  non_responders <- meta %>% dplyr::filter(study_id %in% c(2, 19)) %>% row.names()
}

if (!rebuild_rds & file.exists(dds_file)){
    dds <- readRDS(dds_file)
}else{
    dds <- DESeqDataSetFromTximport(txi,
                                    colData = meta,
                                    design = ~response)

    if (remove_cases_2_19){
        dds <- dds[,!colnames(dds) %in% non_responders]
    }
    design(dds) <- formula(~response + er + tumor_percentage_high + date_of)

    # subset protein-coding genes
    pc_genes <- intersect(protein_coding_genes$ensembl_gene_id, row.names(dds))
    dds <- dds[pc_genes,]
    # 100 reads / 20 samples
    keep <- rowSums(counts(dds)) >= 100
    dds <- dds[keep,]

    # Run DESeq2
    dds <- DESeq(dds)
    saveRDS(dds, dds_file)
}
```

DEGreport QC

Size factor QC - samples 1-20

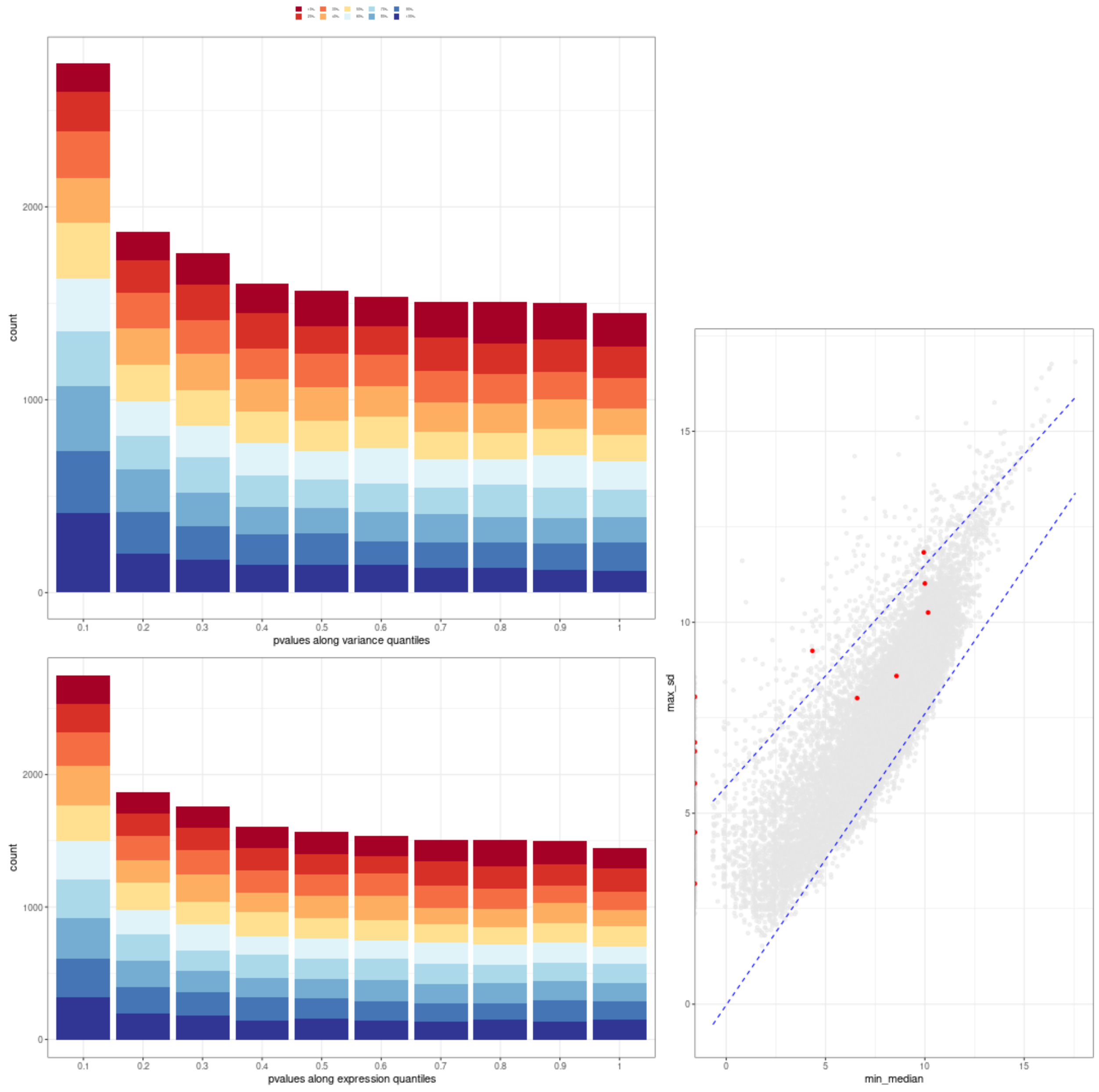
```
counts <- counts(dds, normalized = TRUE)
design <- as.data.frame(colData(dds))
degCheckFactors(counts[, 1:20])
```



Mean-Variance QC plots

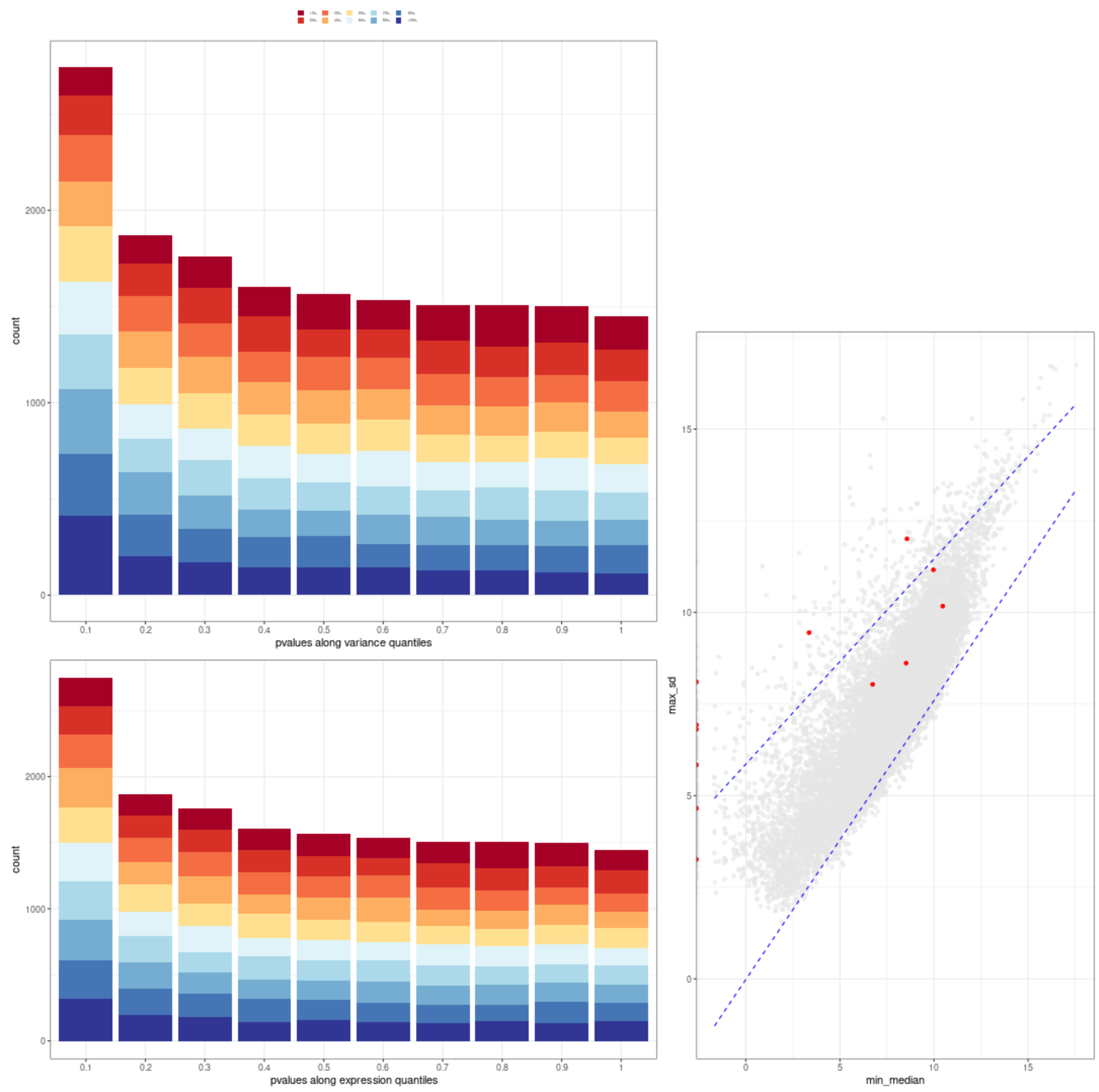
response

```
res <- results(dds)
degQC(counts, design[["response"]], pvalue = res[["pvalue"]])
```



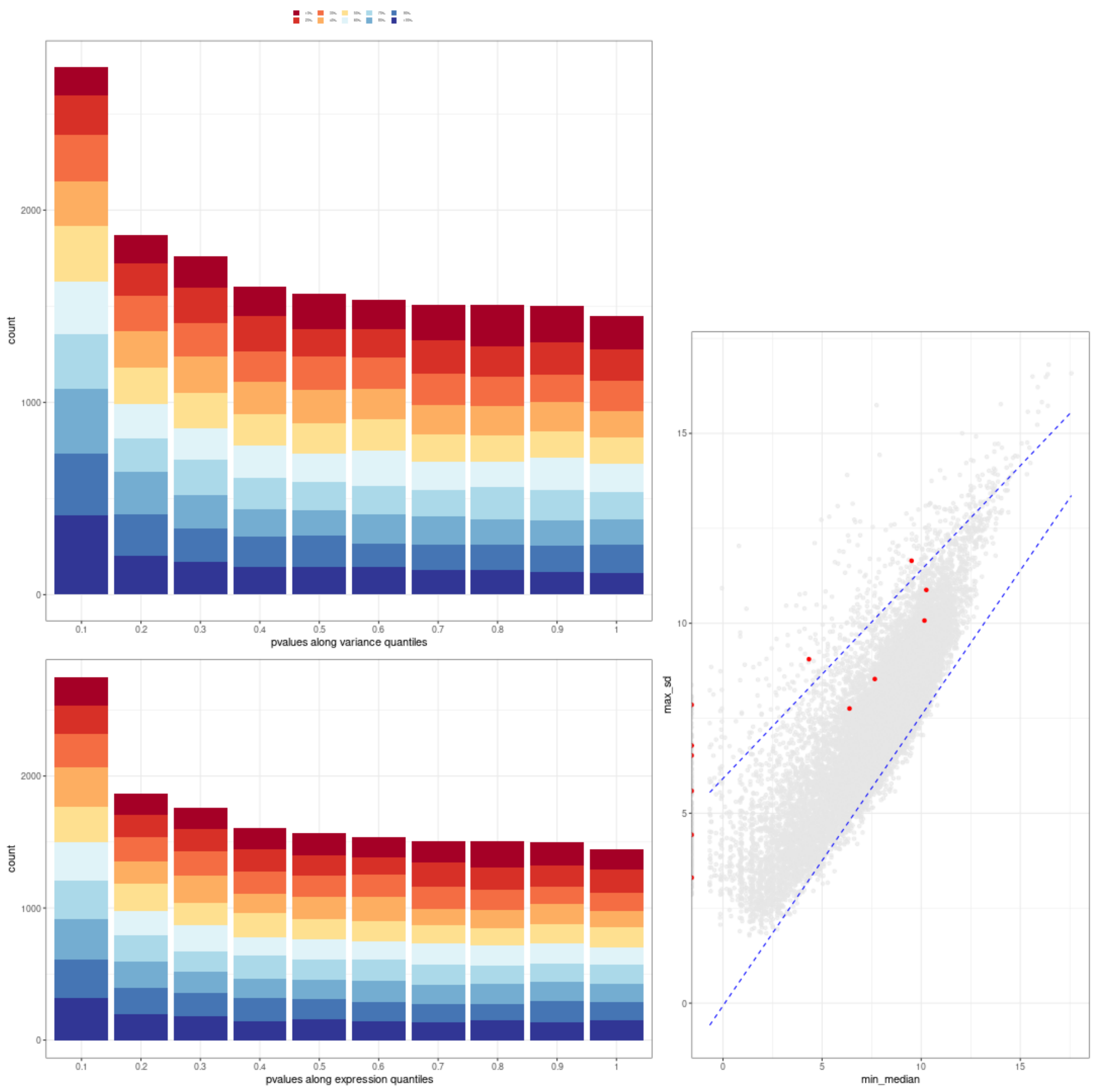
ER

```
degQC(counts, design[["er"]], pvalue = res[["pvalue"]])
```



tumor_percentage_high

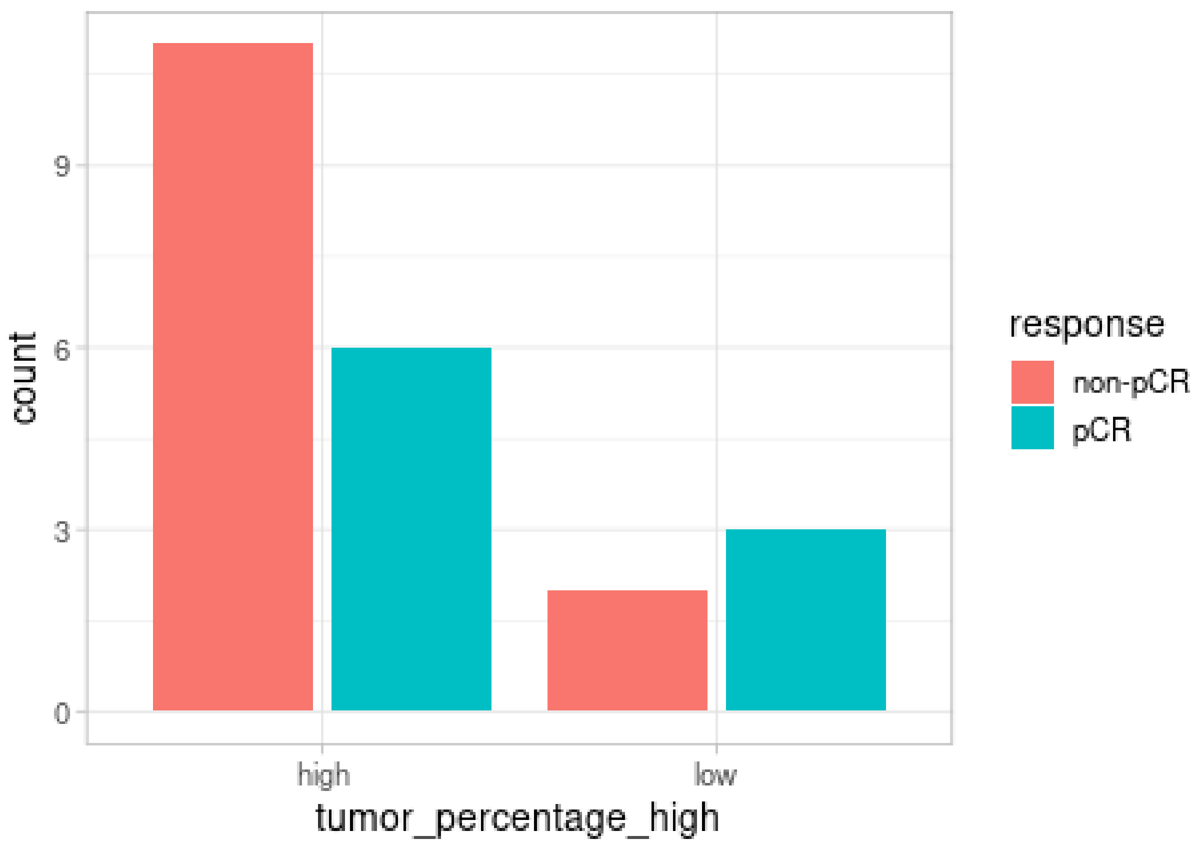
```r
degQC(counts, design[["tumor_percentage_high"]], pvalue = res[["pvalue"]])
```



Covariates effect on count data

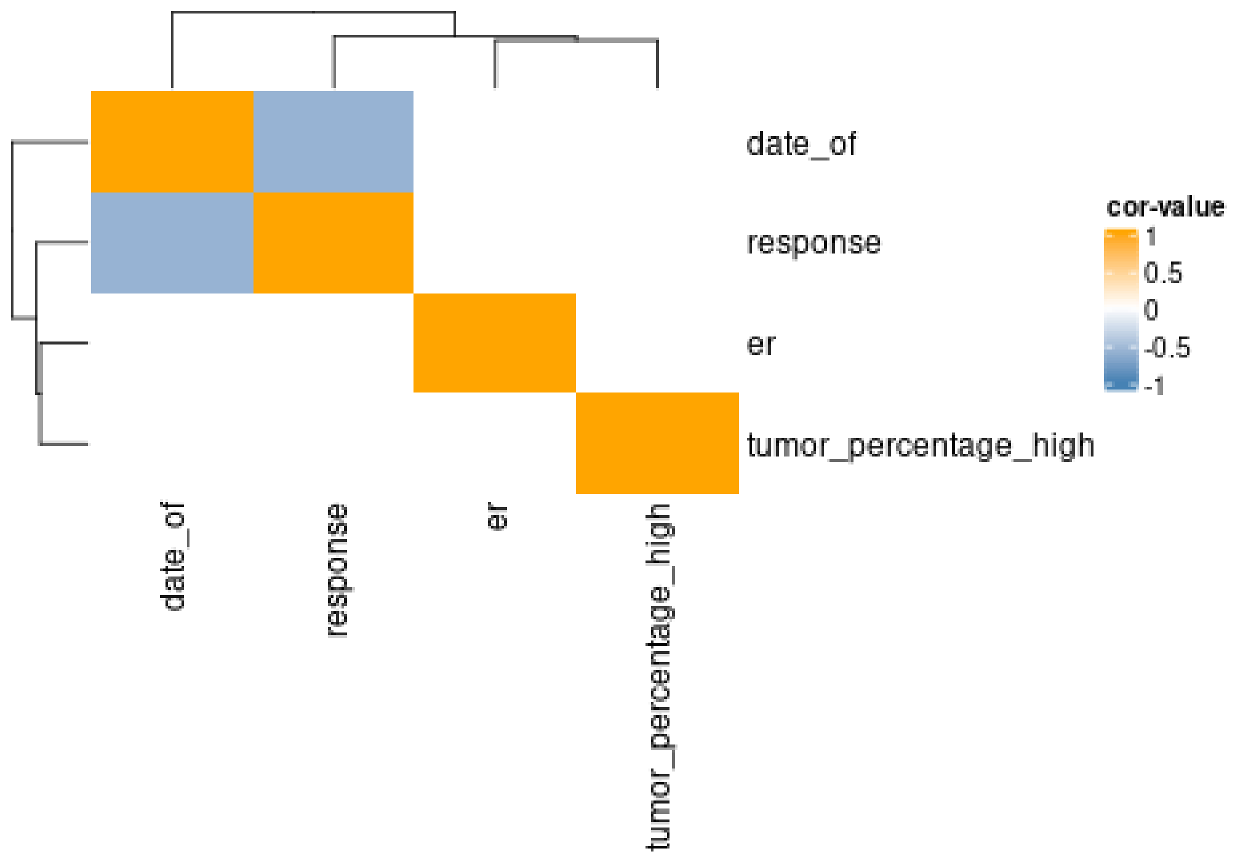
```
mdata <- colData(dds) %>%  as.data.frame()  %>%
  dplyr::select(response, er, date_of, tumor_percentage_high)

#resCov <- degCovariates(log2(counts(dds)+0.5), mdata)
```

```
mdata %>% ggplot(aes(tumor_percentage_high, fill = response)) + geom_bar(position = "dodge2")
```

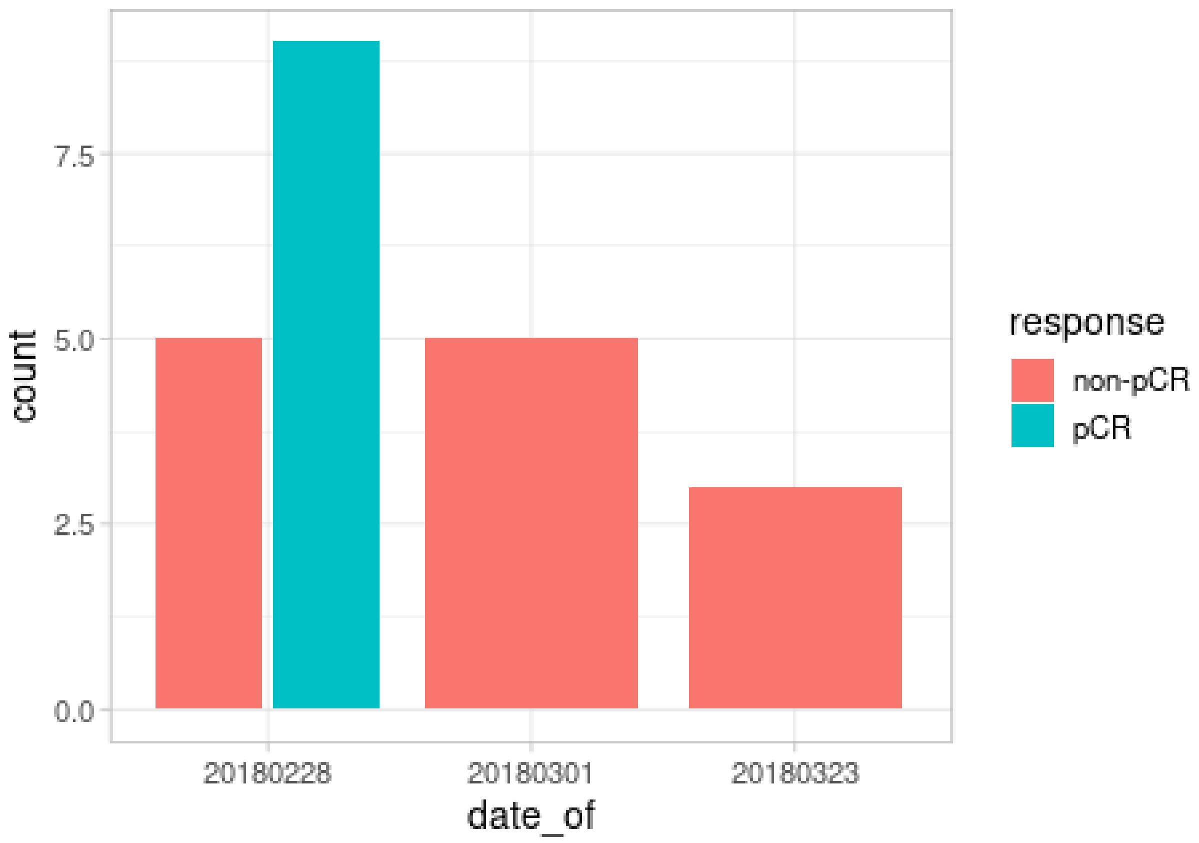


Covariates correlation with metrics

```r
cor <- degCorCov(mdata)
```



```
mdata %>% ggplot(aes(date_of, fill = response)) + geom_bar(position = "dodge2")
```



Sample-level QC analysis

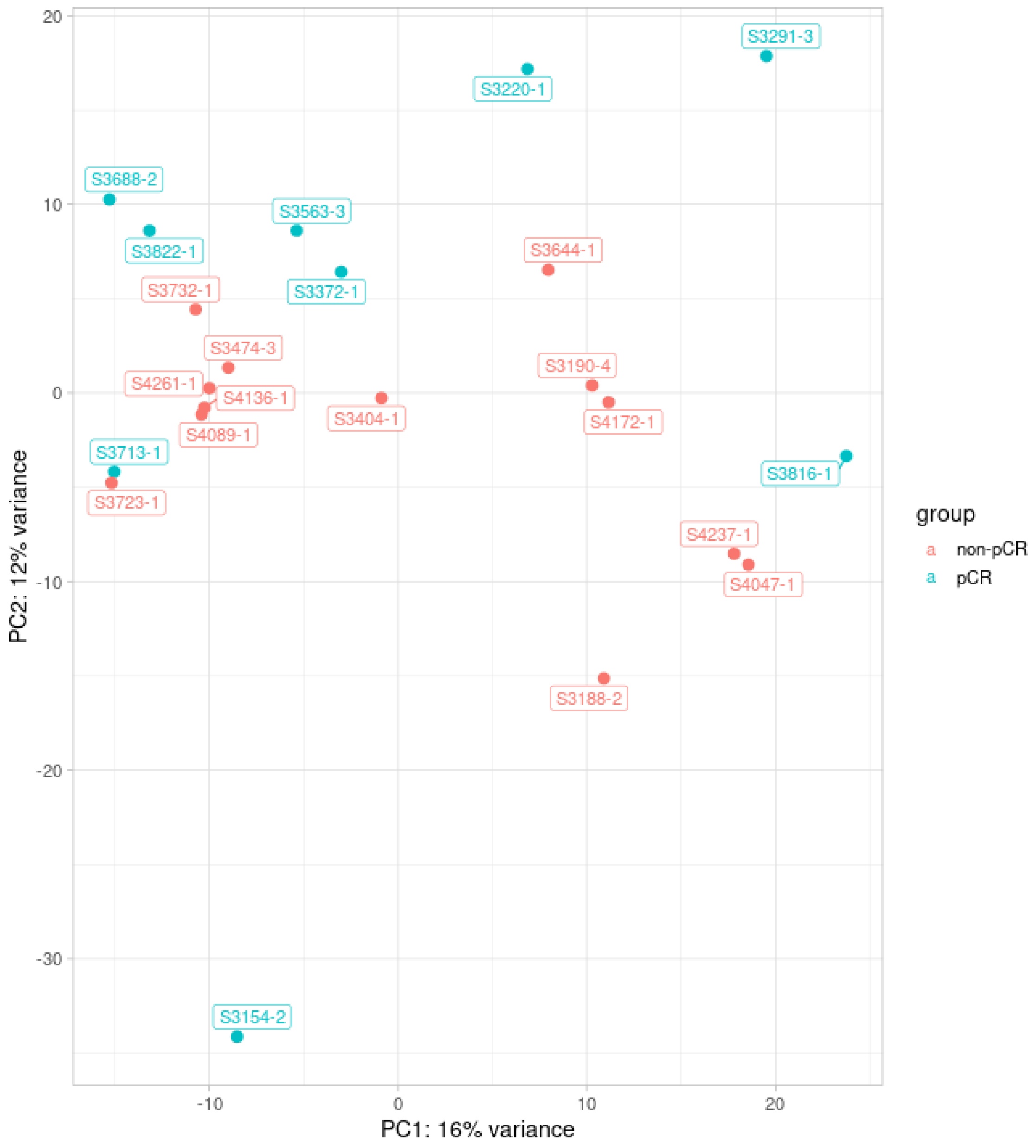
```
### Transform counts for data visualization (unsupervised analysis)
rld_file <- "data/rld.day1.RDS"
if (!rebuild_rds &file.exists(rld_file)){
    rld <- readRDS(rld_file)
}else{
    rld <- rlog(dds, blind = TRUE)
    saveRDS(rld, rld_file)
}
class(rld) # what type of object is this
```

```
## [1] "DESeqTransform"
## attr(,"package")
## [1] "DESeq2"
```

```
# we also need just a matrix of transformed counts
rld_mat <- assay(rld)
```

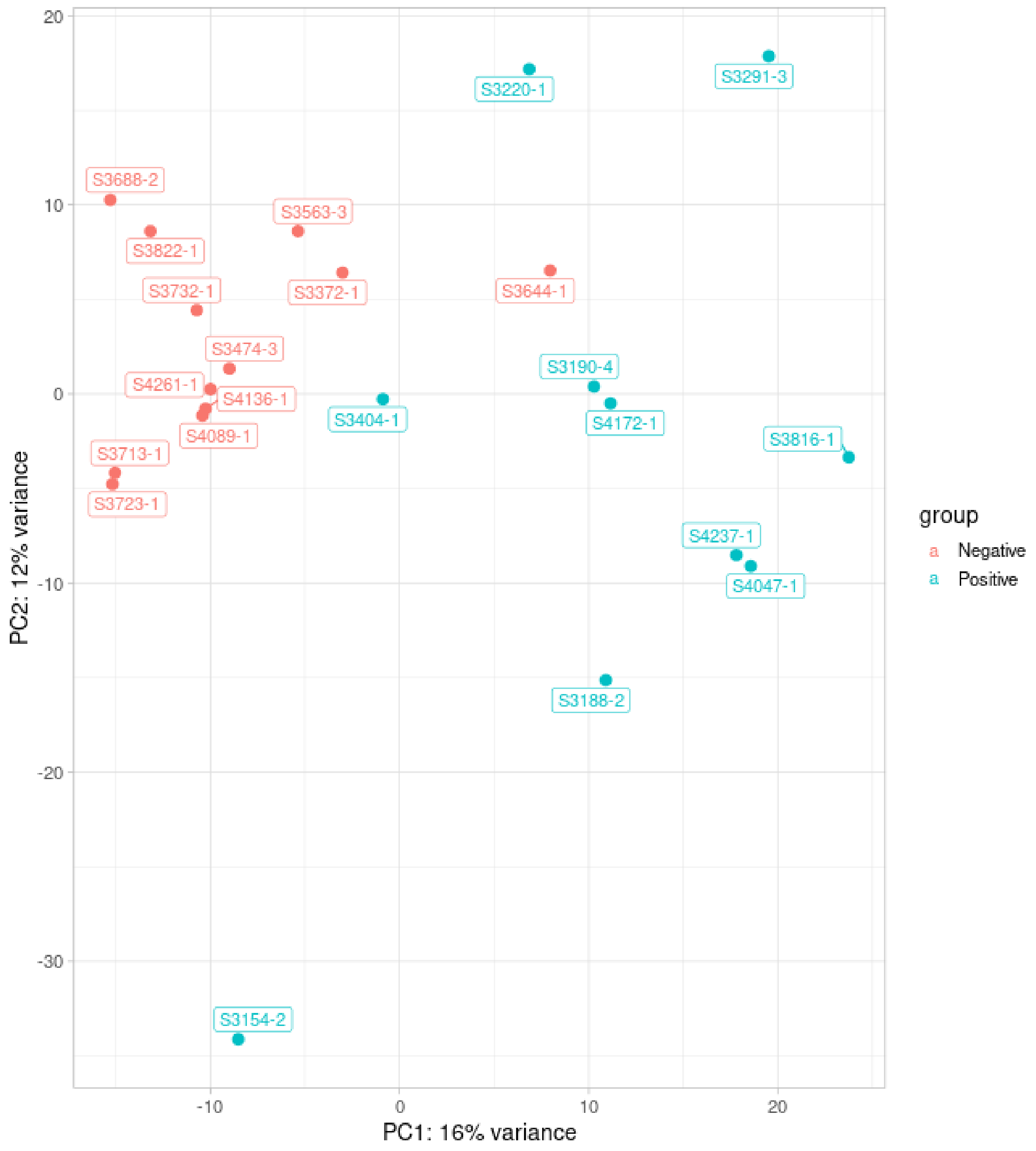
PCA - response

```r
# Use the DESeq2 function
plotPCA(rld, intgroup = c("response")) + geom_label_repel(aes(label = name))
```



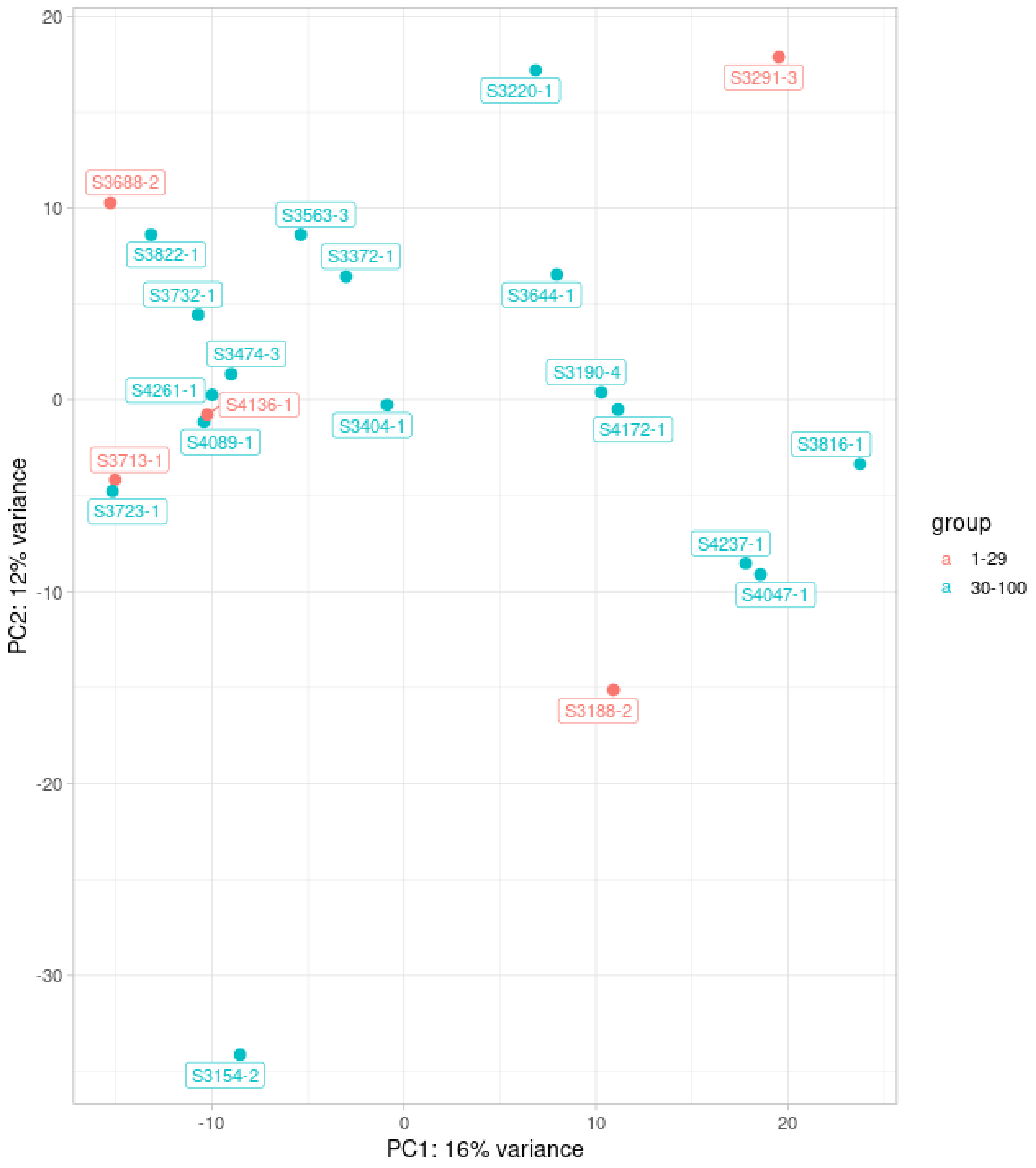
PCA - ER

```r
# Use the DESeq2 function
plotPCA(rld, intgroup = c("er")) + geom_label_repel(aes(label = name))
```



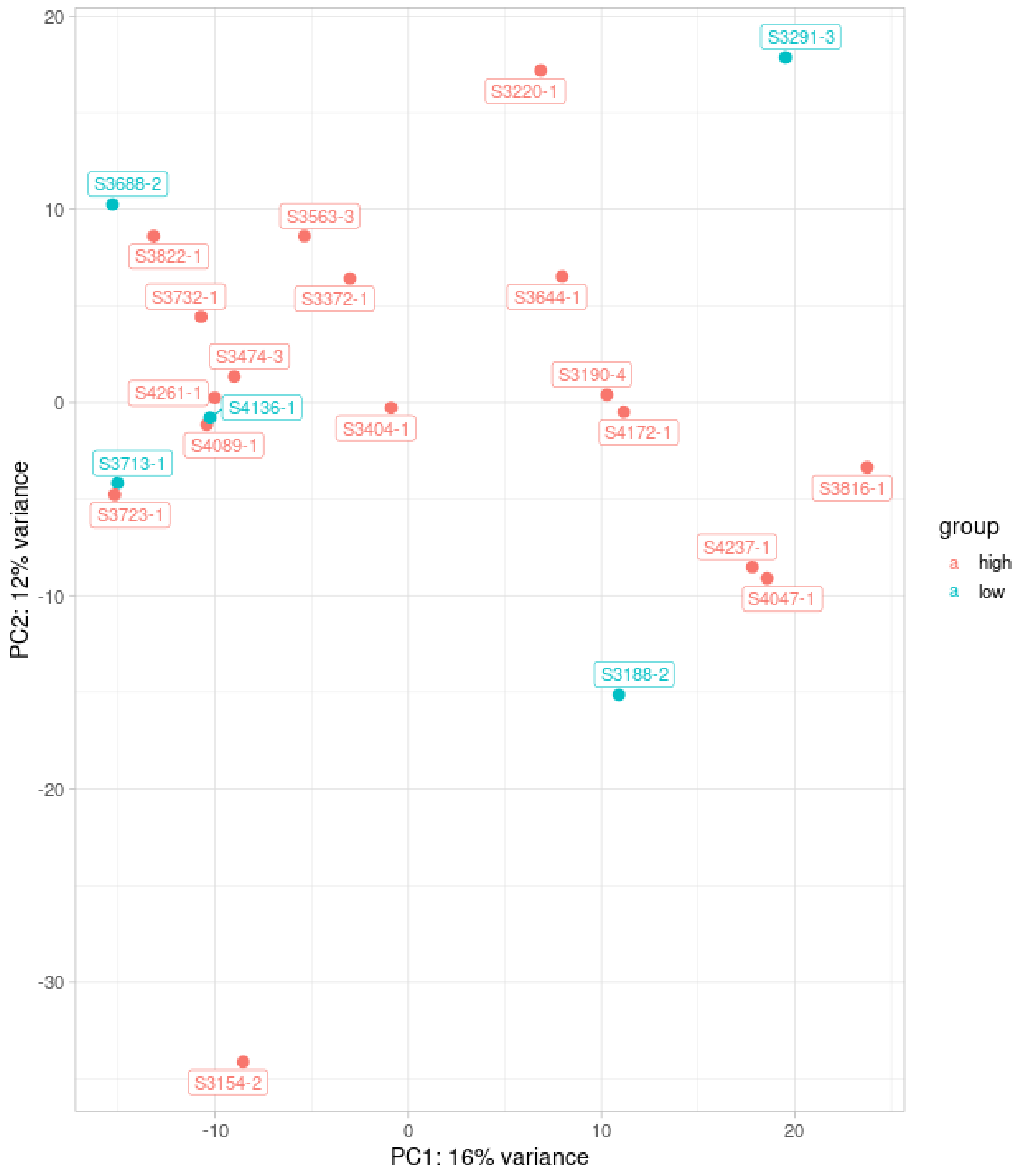
PCA - tumor_percentage

```r
# Use the DESeq2 function
plotPCA(rld, intgroup = c("tumor_percentage")) + geom_label_repel(aes(label = name))
```



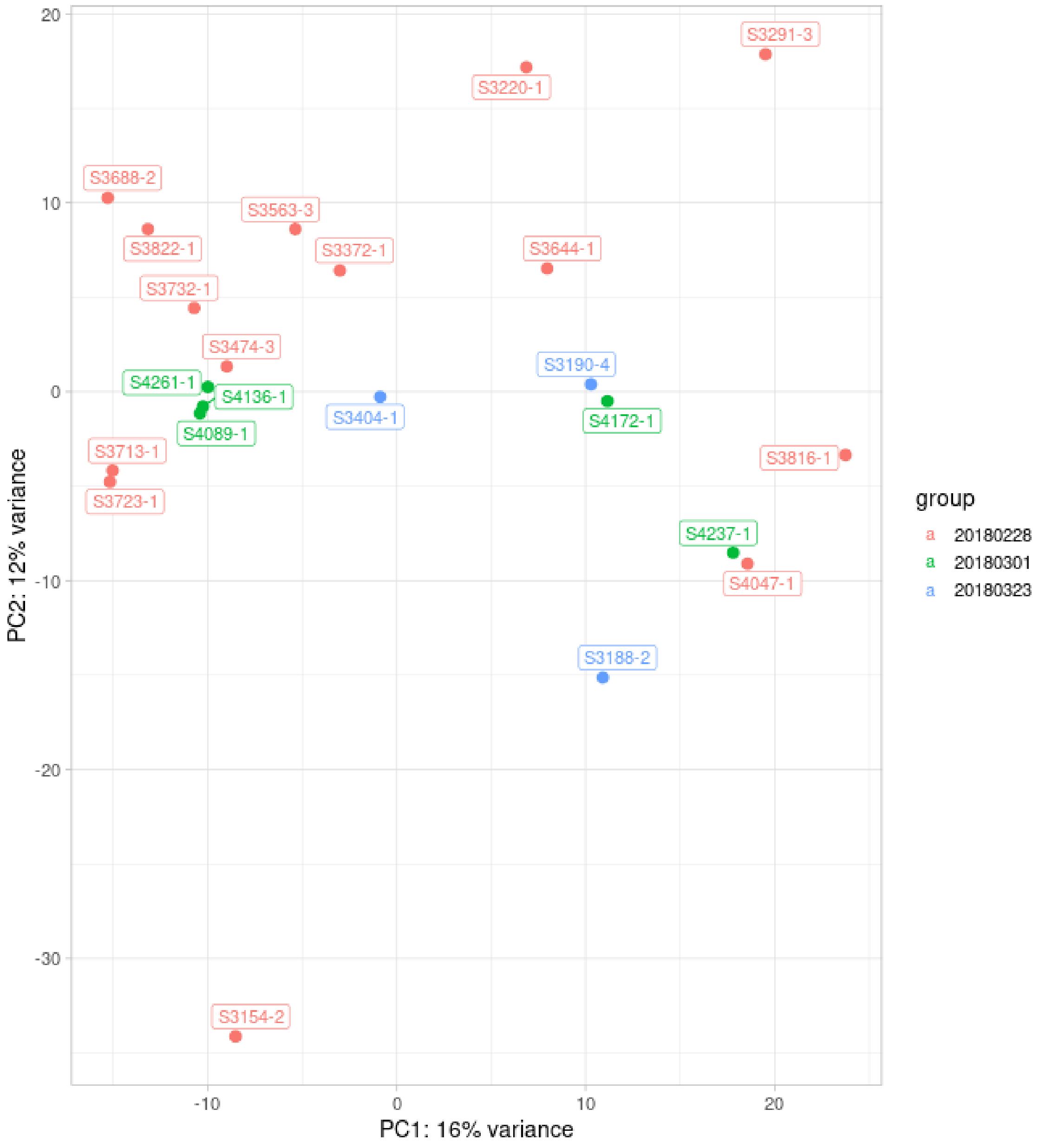
PCA - tumor_percentage_high

```r
# Use the DESeq2 function
plotPCA(rld, intgroup = c("tumor_percentage_high")) + geom_label_repel(aes(label = name))
```



PCA - date_of

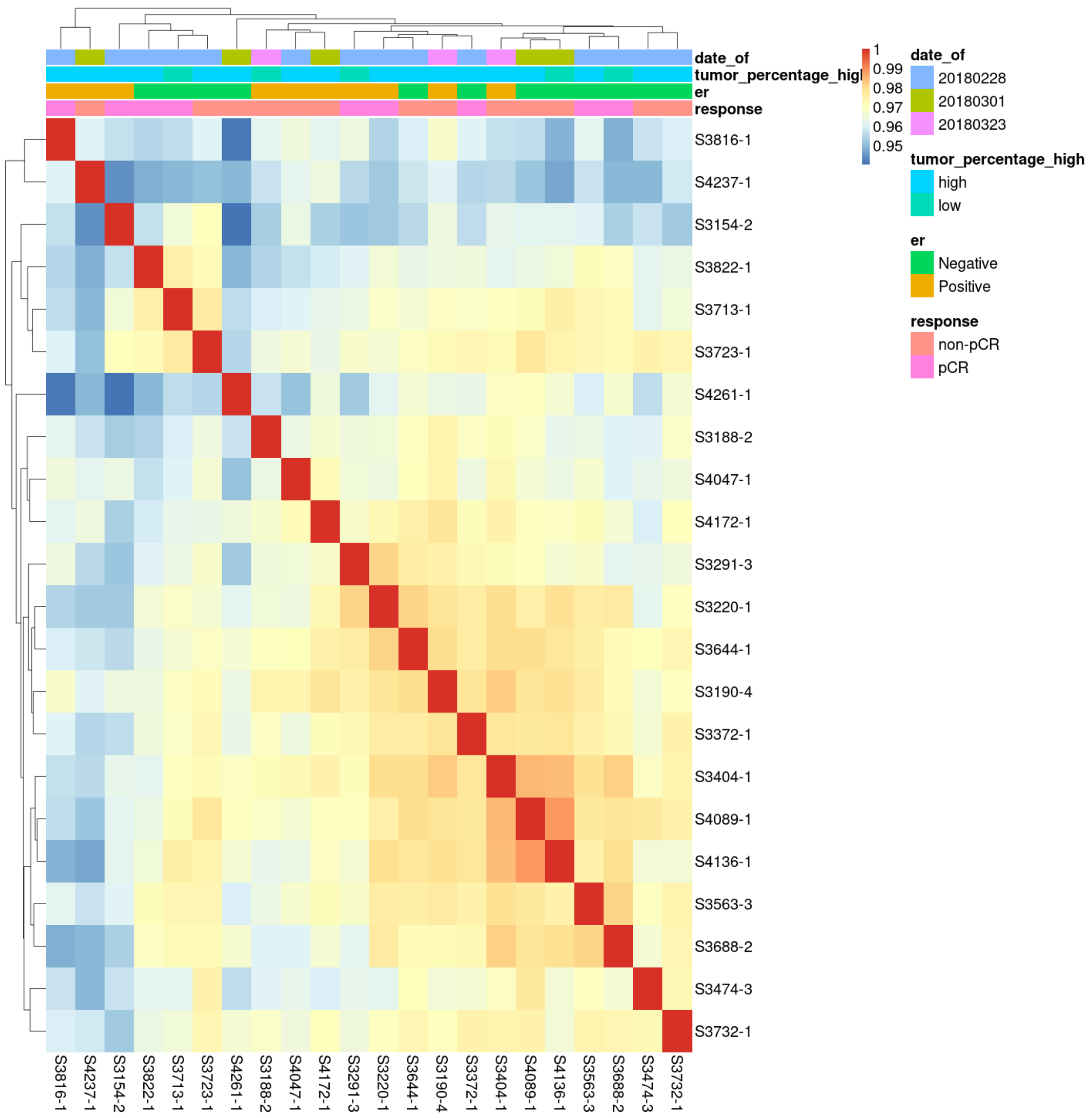
```
# Use the DESeq2 function
plotPCA(rld, intgroup = c("date_of")) + geom_label_repel(aes(label = name))
```



Inter-correlation analysis

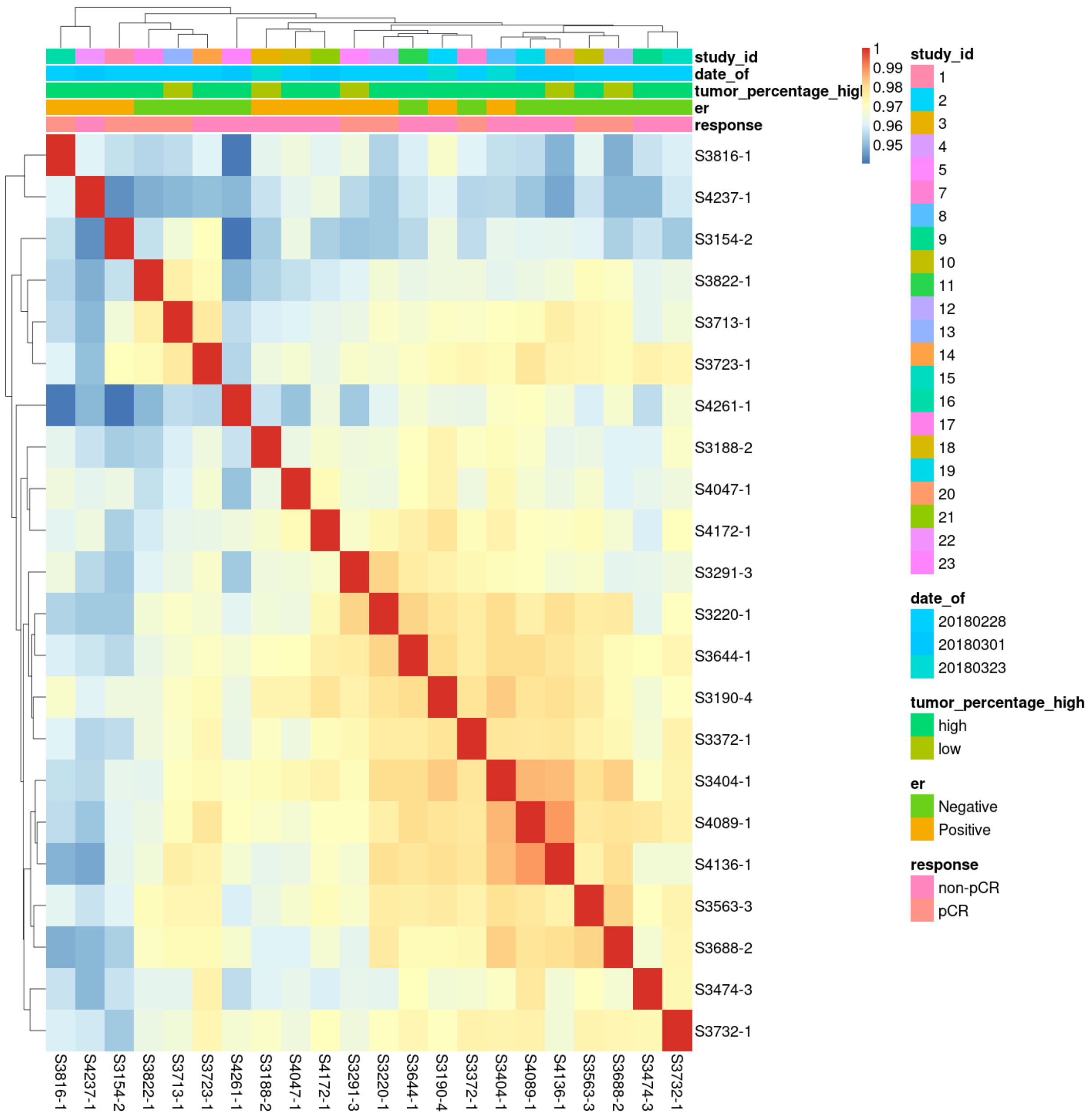
Without study_id

```r
# Correlation matrix
rld_cor <- cor(rld_mat)

meta$study_id <- as.factor(meta$study_id)
# Create annotation file for samples
annotation <- meta[, c("response", "er", "tumor_percentage_high", "date_of")]

# Change colors
heat.colors <- brewer.pal(6, "Blues")

# Plot heatmap
pheatmap(rld_cor,
         annotation = annotation,
         border = NA,
         fontsize = 20)
```


date_of
tumor_percentage_high
er
response
1
0.99
0.98
0.97
0.96
0.95
date_of
20180228
20180301
20180323
tumor_percentage_high
high
low
er
Negative
Positive
response
non-pCR
pCR
S3816-1
S4237-1
S3154-2
S3822-1
S3713-1
S3723-1
S4261-1
S3188-2
S4047-1
S4172-1
S3291-3
S3220-1
S3644-1
S3190-4
S3372-1
S3404-1
S4089-1
S4136-1
S3563-3
S3688-2
S3474-3
S3732-1

With study_id

```r
# Correlation matrix
rld_cor <- cor(rld_mat)

meta$study_id <- as.factor(meta$study_id)
# Create annotation file for samples
annotation <- meta[, c("response", "er", "tumor_percentage_high", "date_of", "study_id")]

# Change colors
heat.colors <- brewer.pal(6, "Blues")

# Plot heatmap
pheatmap(rld_cor,
         annotation = annotation,
         border = NA,
         fontsize = 20)
```


study_id
date_of
tumor_percentage_high
er
response
S3816-1
S4237-1
S3154-2
S3822-1
S3713-1
S3723-1
S4261-1
S3188-2
S4047-1
S4172-1
S3291-3
S3220-1
S3644-1
S3190-4
S3372-1
S3404-1
S4089-1
S4136-1
S3563-3
S3688-2
S3474-3
S3732-1
1
0.99
0.98
0.97
0.96
0.95
study_id
1
2
3
4
5
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
date_of
20180228
20180301
20180323
tumor_percentage_high
high
low
er
Negative
Positive
response
non-pCR
pCR

Response pCR vs non-pCR for Day 1- see Table9

ER : Positive vs Negative for Day1 - Table 10

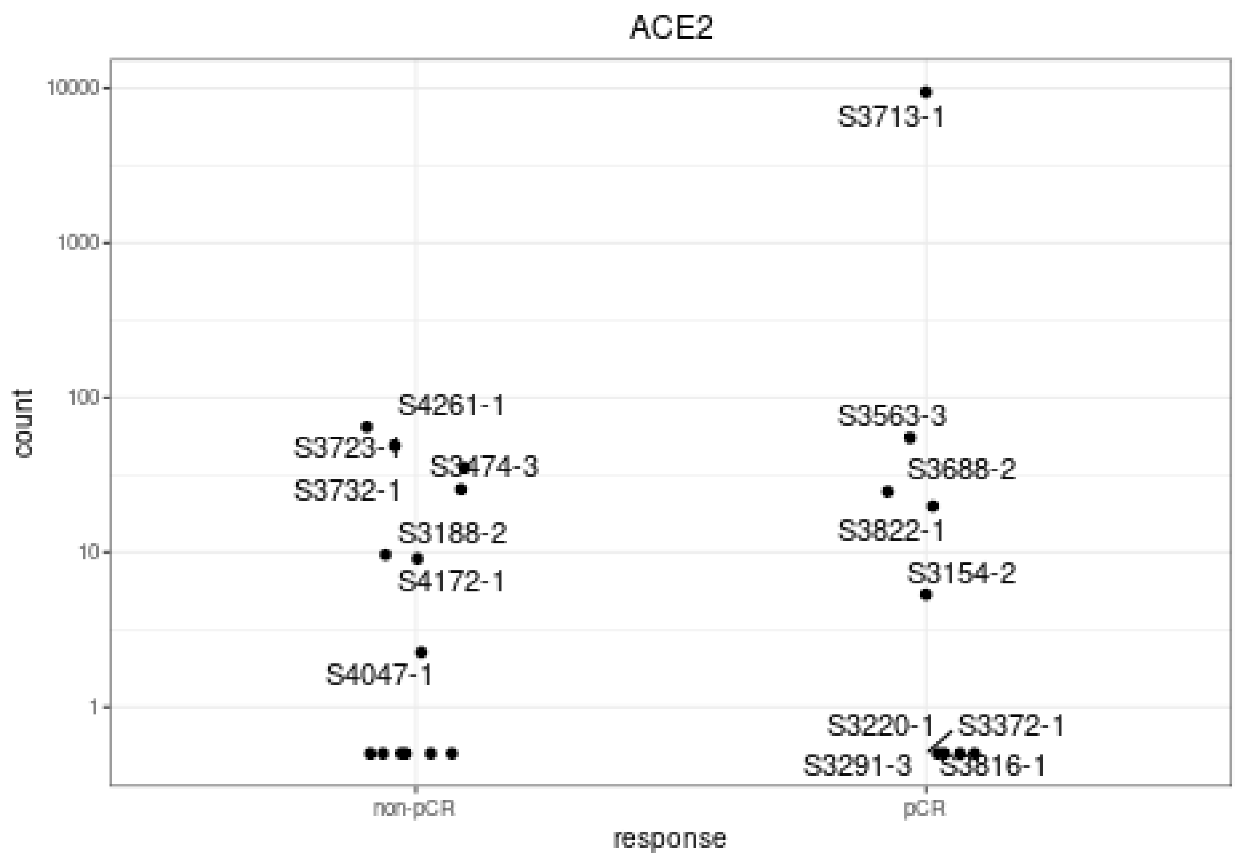
tumor_percentage_high : High vs Low for Day1- Table 11

date_of: 20180323 vs 20180228 - for Day1: Table 12

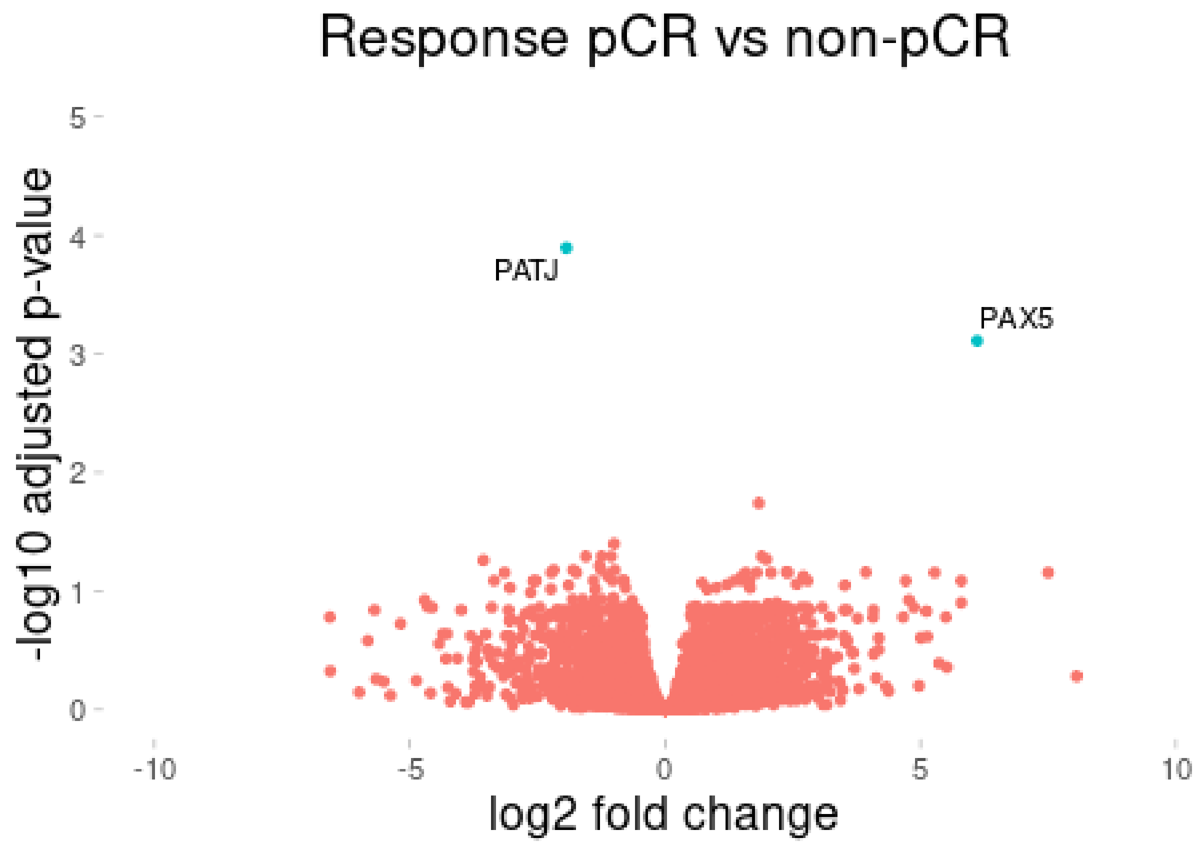
Visualization

Gene example

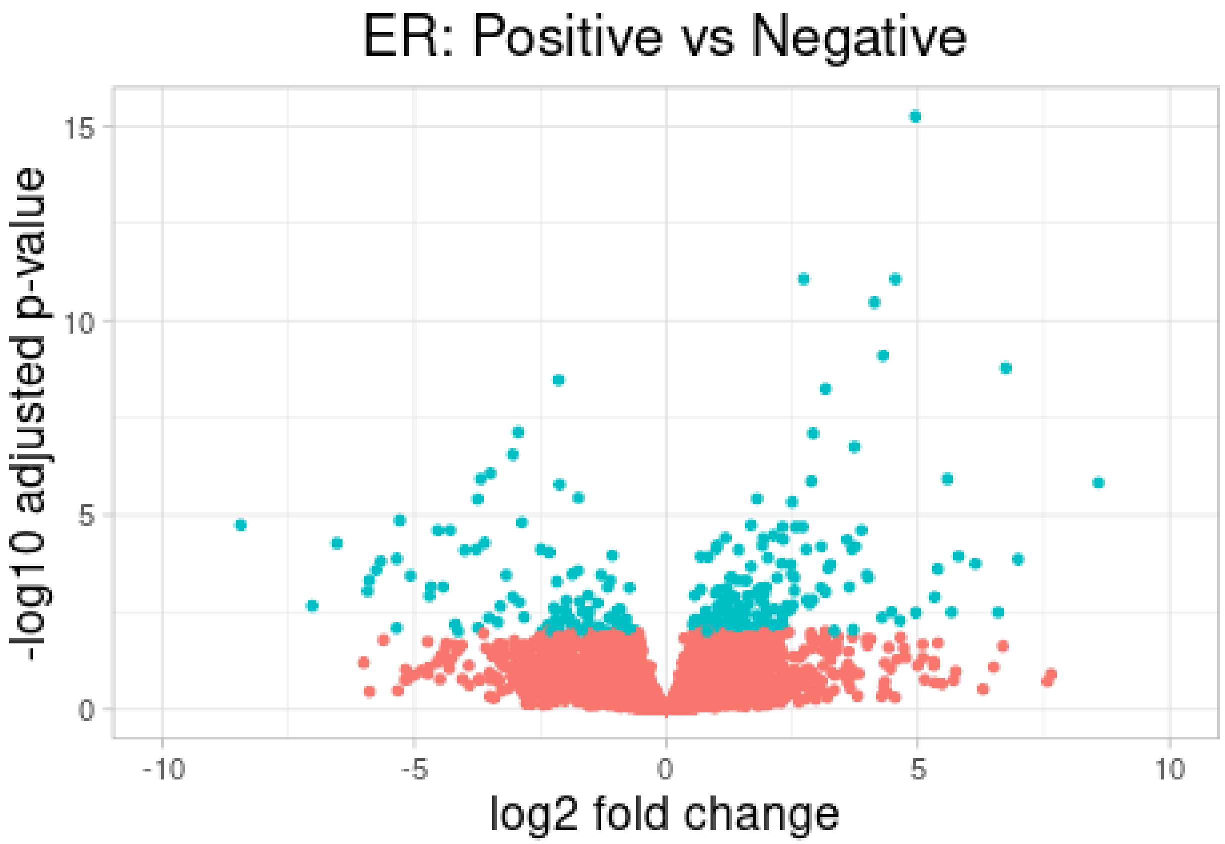
```r
d <- plotCounts(dds,
                gene = "ENSG00000130234",
                intgroup = "response",
                returnData = TRUE)

ggplot(d, aes(x = response, y = count)) +
     geom_point(position = position_jitter(w = 0.1, h = 0)) +
     geom_text_repel(aes(label = rownames(d))) +
     theme_bw(base_size = 10) +
     ggtitle("ACE2") +
     theme(plot.title = element_text(hjust = 0.5)) +
     scale_y_log10()
```



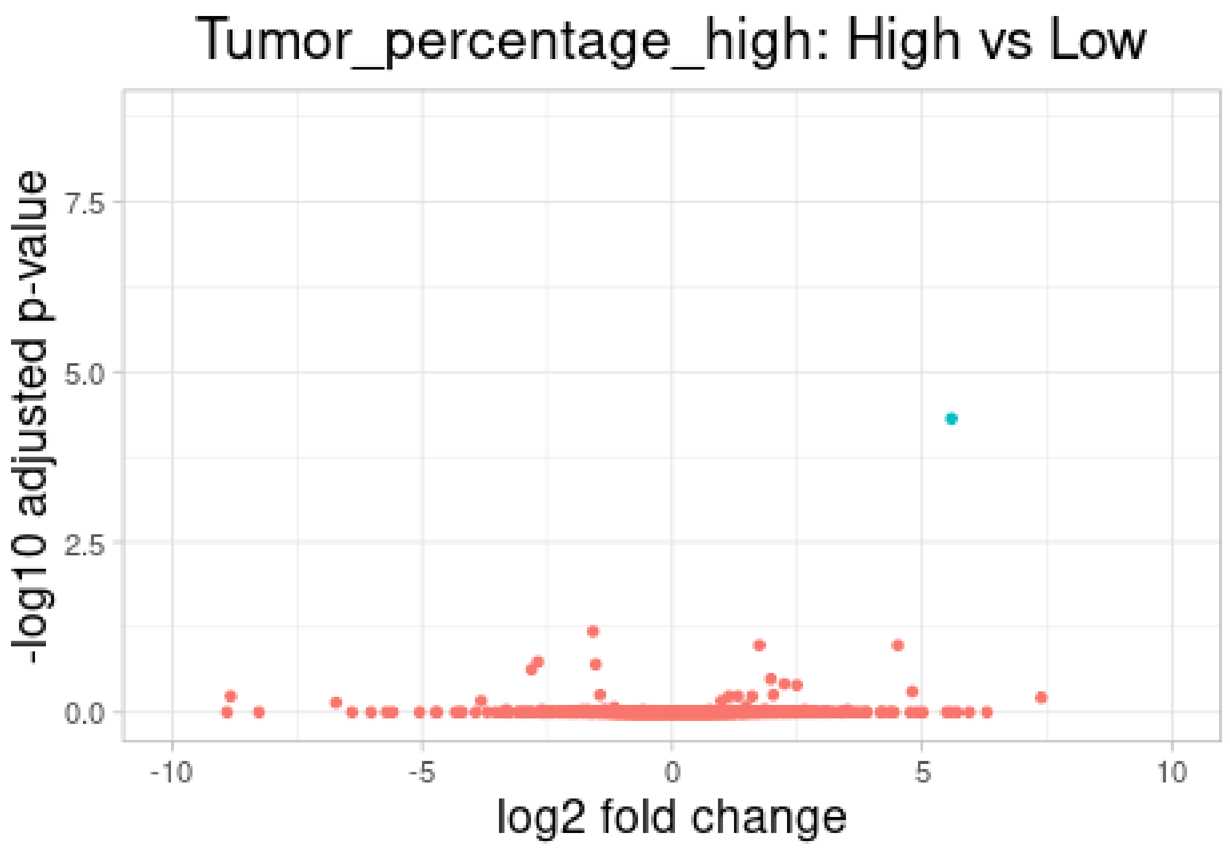
```r
# Add a column for significant genes
resResponse_tb_vis <- resResponse_tb %>% mutate(threshold = padj < 0.01)

resResponse_tb_vis$symbol <- ifelse((abs(resResponse_tb_vis$log2FoldChange) > 1.5),
                                resResponse_tb_vis$symbol, NA)

resResponse_tb_vis$symbol <- ifelse(resResponse_tb_vis$threshold,
                                    resResponse_tb_vis$symbol, NA)

ggplot(resResponse_tb_vis,
       aes(log2FoldChange, -log10(padj), label = symbol)) +
  geom_point(aes(x = log2FoldChange, y = -log10(padj), colour = threshold)) +
  ggtitle("Response pCR vs non-pCR") +
  xlab("log2 fold change") +
  ylab("-log10 adjusted p-value") +
  scale_x_continuous(limits = c(-10,10)) +
  scale_y_continuous(limits = c(0, 5))+
  theme(legend.position = "none",
        plot.title = element_text(size = rel(1.5), hjust = 0.5),
        axis.title = element_text(size = rel(1.25)),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank()) +
  geom_text_repel(aes(label = symbol))
```



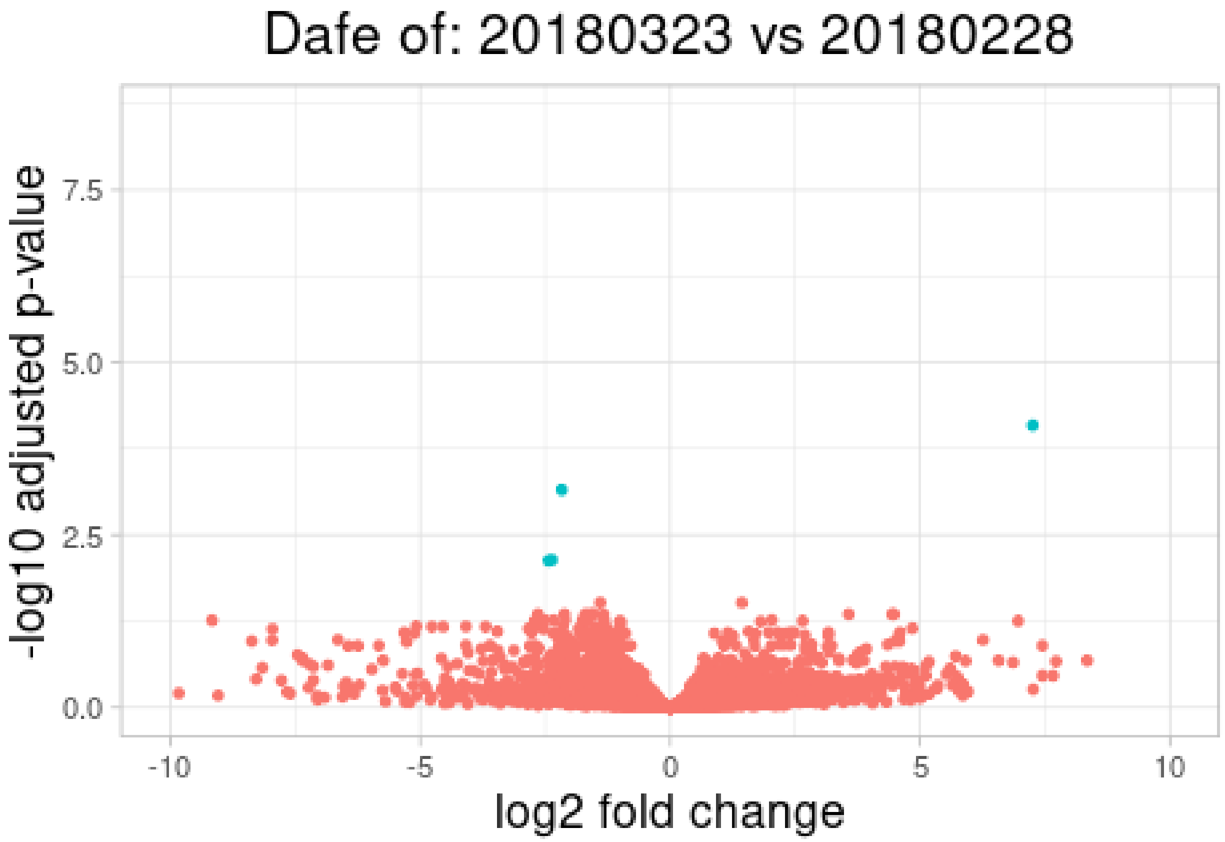
```r
# Add a column for significant genes
resER_tb <- resER_tb %>% mutate(threshold = padj < 0.01)

ggplot(resER_tb) +
  geom_point(aes(x = log2FoldChange, y = -log10(padj), colour = threshold)) +
  ggtitle("ER: Positive vs Negative") +
  xlab("log2 fold change") +
  ylab("-log10 adjusted p-value") +
  scale_x_continuous(limits = c(-10,10)) +
  theme(legend.position = "none",
        plot.title = element_text(size = rel(1.5), hjust = 0.5),
        axis.title = element_text(size = rel(1.25)))
```



```r
# Add a column for significant genes
resTP_tb <- resTP_tb %>% mutate(threshold = padj < 0.01)

ggplot(resTP_tb) +
  geom_point(aes(x = log2FoldChange, y = -log10(padj), colour = threshold)) +
  ggtitle("Tumor_percentage_high: High vs Low") +
  xlab("log2 fold change") +
  ylab("-log10 adjusted p-value") +
  scale_x_continuous(limits = c(-10,10)) +
  theme(legend.position = "none",
        plot.title = element_text(size = rel(1.5), hjust = 0.5),
        axis.title = element_text(size = rel(1.25)))
```

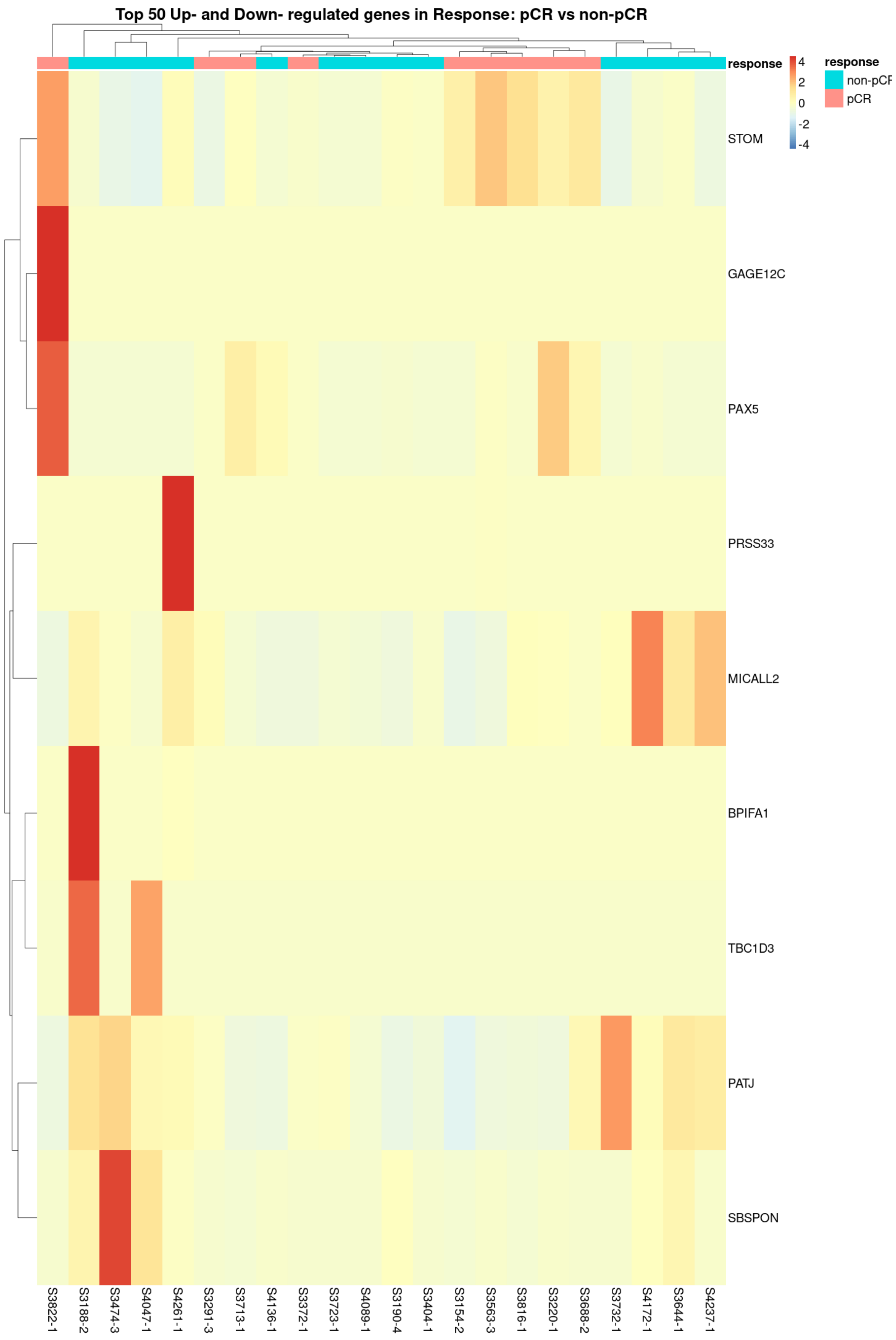


```r
# Add a column for significant genes
resDO_tb <- resDO_tb %>% mutate(threshold = padj < 0.01)

ggplot(resDO_tb) +
  geom_point(aes(x = log2FoldChange, y = -log10(padj), colour = threshold)) +
  ggtitle("Dafe of: 20180323 vs 20180228") +
  xlab("log2 fold change") +
  ylab("-log10 adjusted p-value") +
  scale_x_continuous(limits = c(-10,10)) +
  theme(legend.position = "none",
        plot.title = element_text(size = rel(1.5), hjust = 0.5),
        axis.title = element_text(size = rel(1.25)))
```

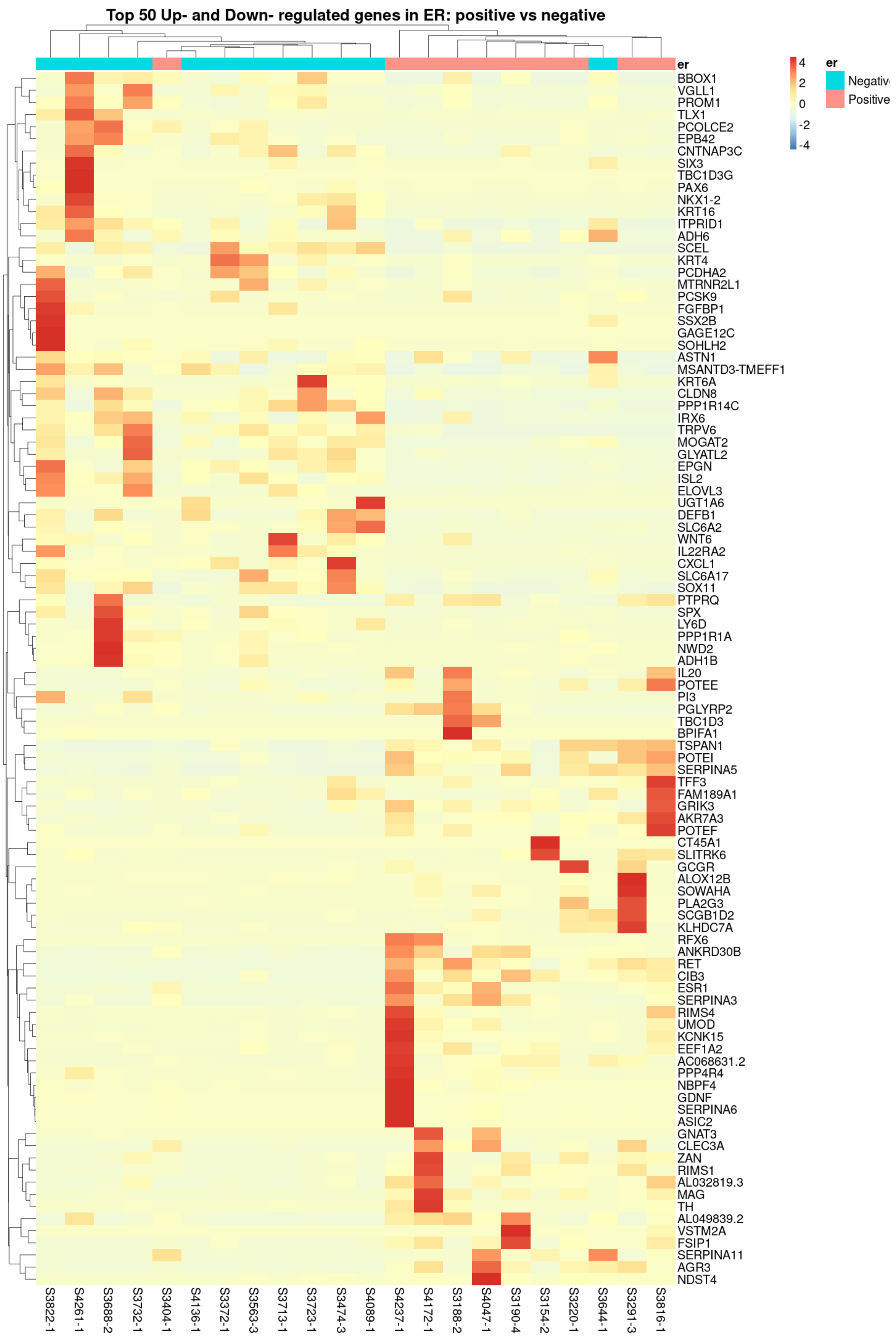


Heatmaps

```
# Create a matrix of normalized expression
sig_up <- resResponse_tb_significant %>% arrange(-log2FoldChange) %>% head(50) %>% pull(gene)
sig_down <- resResponse_tb_significant %>% arrange(log2FoldChange) %>% head(50) %>% pull(gene)
sig <- c(sig_up, sig_down)

row_annotation <- gene_symbol %>%
                    as_tibble() %>%
                    dplyr::filter(gene_id %in% sig)

plotmat <- txi$abundance[c(sig_up, sig_down),] %>% as.data.frame() %>%
          rownames_to_column(var = "ensembl_gene_id") %>%
          left_join(gene_symbol, by = c("ensembl_gene_id" = "gene_id")) %>%
          drop_na(symbol)

plotmat$ensembl_gene_id <- NULL

plotmat <- plotmat %>% column_to_rownames(var = "symbol") %>% as.matrix()

# Color palette
heat.colors <- brewer.pal(6, "YlOrRd")

# Plot heatmap
pheatmap(plotmat,
         scale = "row",
         show_rownames = TRUE,
         border = FALSE,
         annotation = meta[, c("response"), drop = FALSE],
         main = "Top 50 Up- and Down- regulated genes in Response: pCR vs non-pCR",
         fontsize = 20)
```

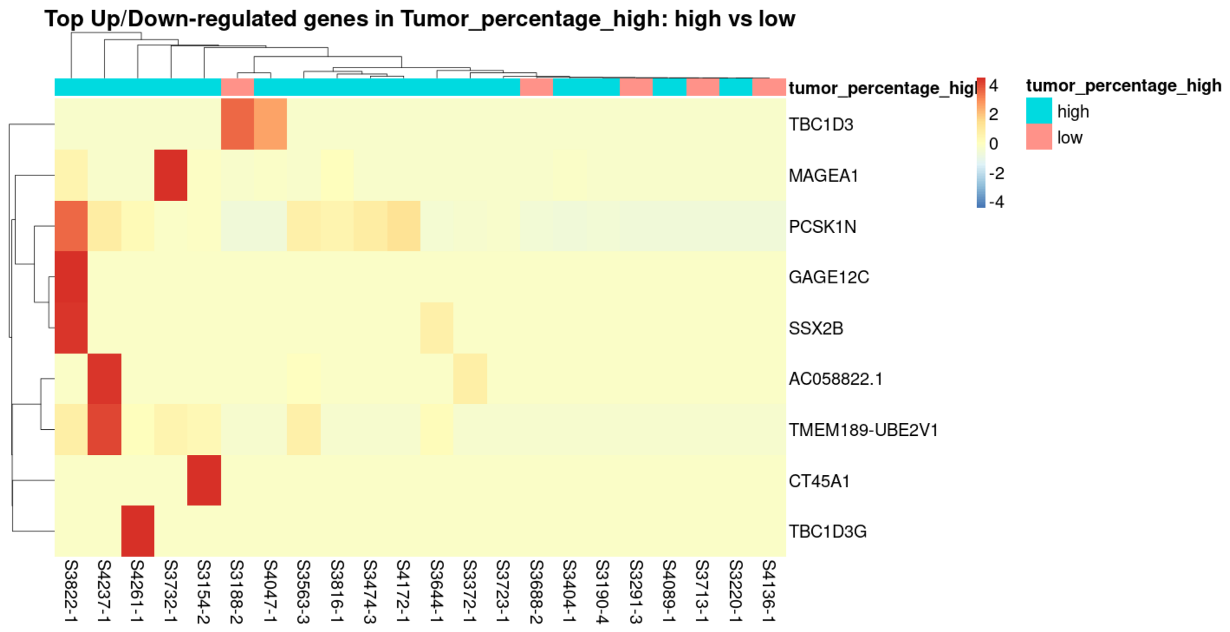

Top 50 Up- and Down- regulated genes in Response: pCR vs non-pCR
response
STOM
GAGE12C
PAX5
PRSS33
MICALL2
BPIFA1
TBC1D3
PATJ
SBSPON
4
2
0
-2
-4
response
non-pCF
pCR
S3822-1
S3188-2
S3474-3
S4047-1
S4261-1
S3291-3
S3713-1
S4136-1
S3372-1
S3723-1
S4089-1
S3190-4
S3404-1
S3154-2
S3563-3
S3816-1
S3220-1
S3688-2
S3732-1
S4172-1
S3644-1
S4237-1

```r
# Create a matrix of normalized expression
sig_up <- resER_tb_significant %>% arrange(-log2FoldChange) %>% head(50) %>% pull(gene)
sig_down <- resER_tb_significant %>% arrange(log2FoldChange) %>% head(50) %>% pull(gene)
sig <- c(sig_up, sig_down)

row_annotation <- gene_symbol %>%
                    as_tibble() %>%
                    dplyr::filter(gene_id %in% sig)

plotmat <- txi$abundance[c(sig_up, sig_down),] %>% as.data.frame() %>%
          rownames_to_column(var = "ensembl_gene_id") %>%
          left_join(gene_symbol, by = c("ensembl_gene_id" = "gene_id")) %>%
          drop_na(symbol)

plotmat$ensembl_gene_id <- NULL

plotmat <- plotmat %>% column_to_rownames(var = "symbol") %>% as.matrix()

# Color palette
heat.colors <- brewer.pal(6, "YlOrRd")

# Plot heatmap
pheatmap(plotmat,
         scale = "row",
         show_rownames = TRUE,
         border = FALSE,
         annotation = meta[, c("er"), drop = FALSE],
         main = "Top 50 Up- and Down- regulated genes in ER: positive vs negative",
         fontsize = 20)
```

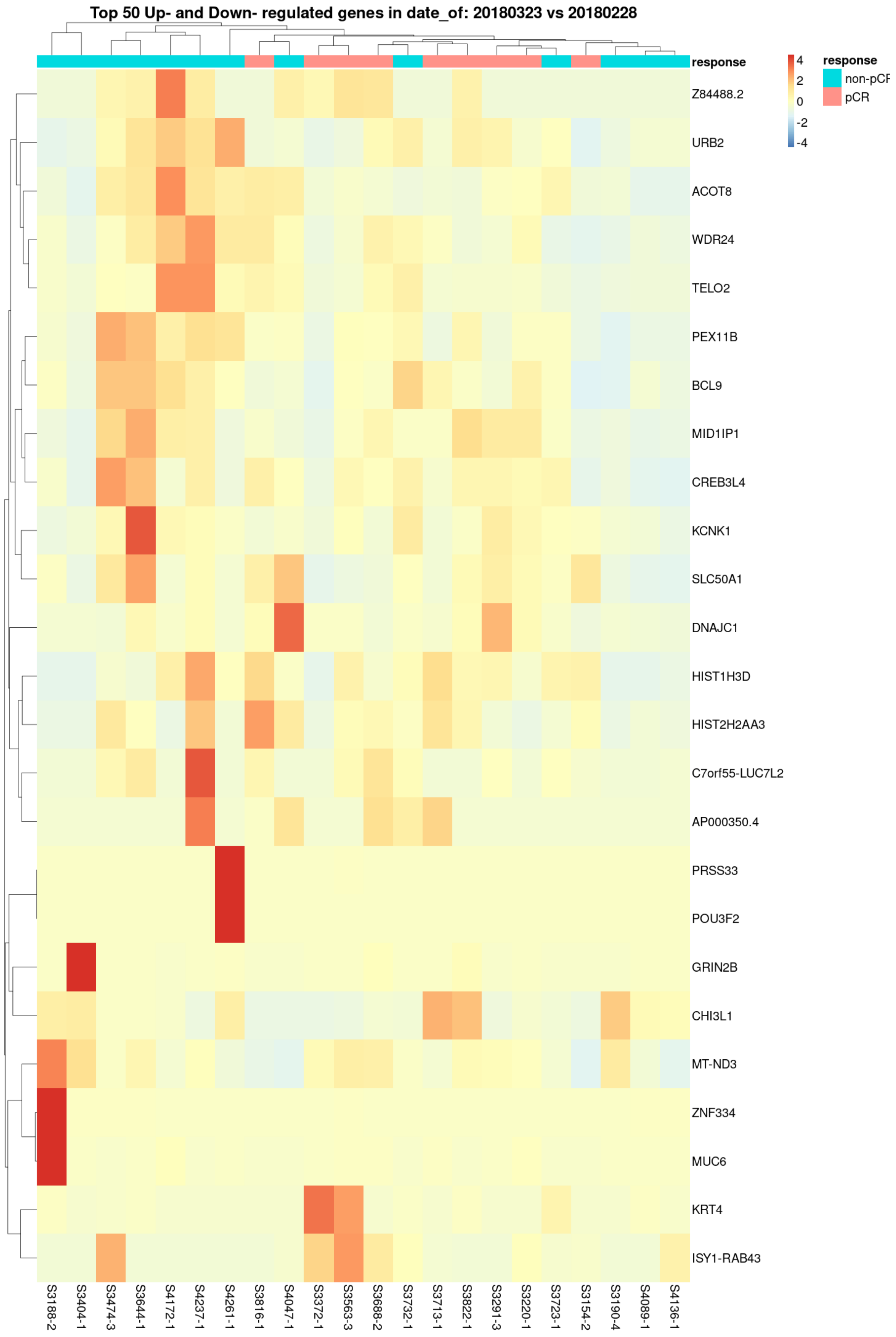


Top 50 Up- and Down- regulated genes in ER: positive vs negative
er
Negativ
Positive
4
2
0
-2
-4
BBOX1
VGLL1
PROM1
TLX1
PCOLCE2
EPB42
CNTNAP3C
SIX3
TBC1D3G
PAX6
NKX1-2
KRT16
ITPRID1
ADH6
SCEL
KRT4
PCDHA2
MTRNR2L1
PCSK9
FGFBP1
SSX2B
GAGE12C
SOHLH2
ASTN1
MSANTD3-TMEFF1
KRT6A
CLDN8
PPP1R14C
IRX6
TRPV6
MOGAT2
GLYATL2
EPGN
ISL2
ELOVL3
UGT1A6
DEFB1
SLC6A2
WNT6
IL22RA2
CXCL1
SLC6A17
SOX11
PTPRQ
SPX
LY6D
PPP1R1A
NWD2
ADH1B
IL20
POTEE
PI3
PGLYRP2
TBC1D3
BPIFA1
TSPAN1
POTEI
SERPINA5
TFF3
FAM189A1
GRIK3
AKR7A3
POTEF
CT45A1
SLITRK6
GCGR
ALOX12B
SOWAHA
PLA2G3
SCGB1D2
KLHDC7A
RFX6
ANKRD30B
RET
CIB3
ESR1
SERPINA3
RIMS4
UMOD
KCNK15
EEF1A2
AC068631.2
PPP4R4
NBPF4
GDNF
SERPINA6
ASIC2
GNAT3
CLEC3A
ZAN
RIMS1
AL032819.3
MAG
TH
AL049839.2
VSTM2A
FSIP1
SERPINA11
AGR3
NDST4
S3822-1
S4261-1
S3688-2
S3732-1
S3404-1
S4136-1
S3372-1
S3563-3
S3713-1
S3723-1
S3474-3
S4089-1
S4237-1
S4172-1
S3188-2
S4047-1
S3190-4
S3154-2
S3220-1
S3644-1
S3291-3
S3816-1

```
# Create a matrix of normalized expression
sig_up <- resTP_tb_significant %>% arrange(-log2FoldChange) %>% head(50) %>% pull(gene)
sig_down <- resTP_tb_significant %>% arrange(log2FoldChange) %>% head(50) %>% pull(gene)
sig <- c(sig_up, sig_down)

row_annotation <- gene_symbol %>%
                    as_tibble() %>%
                    dplyr::filter(gene_id %in% sig)

plotmat <- txi$abundance[c(sig_up, sig_down),] %>% as.data.frame() %>%
          rownames_to_column(var = "ensembl_gene_id") %>%
          left_join(gene_symbol, by = c("ensembl_gene_id" = "gene_id")) %>%
          drop_na(symbol)

plotmat$ensembl_gene_id <- NULL

plotmat <- plotmat %>% column_to_rownames(var = "symbol") %>% as.matrix()

# Color palette
heat.colors <- brewer.pal(6, "YlOrRd")

# Plot heatmap
pheatmap(plotmat,
         scale = "row",
         show_rownames = TRUE,
         border = FALSE,
         annotation = meta[, c("tumor_percentage_high"), drop = FALSE],
         main = "Top Up/Down-regulated genes in Tumor_percentage_high: high vs low",
         fontsize = 20)
```



```r
# Create a matrix of normalized expression
sig_up <- resDO_tb_significant %>% arrange(-log2FoldChange) %>% head(50) %>% pull(gene)
sig_down <- resDO_tb_significant %>% arrange(log2FoldChange) %>% head(50) %>% pull(gene)
sig <- c(sig_up, sig_down)

row_annotation <- gene_symbol %>%
                    as_tibble() %>%
                    dplyr::filter(gene_id %in% sig)

plotmat <- txi$abundance[c(sig_up, sig_down),] %>% as.data.frame() %>%
          rownames_to_column(var = "ensembl_gene_id") %>%
          left_join(gene_symbol, by = c("ensembl_gene_id" = "gene_id")) %>%
          drop_na(symbol)

plotmat$ensembl_gene_id <- NULL

plotmat <- plotmat %>% column_to_rownames(var = "symbol") %>% as.matrix()

# Color palette
heat.colors <- brewer.pal(6, "YlOrRd")

# Plot heatmap
pheatmap(plotmat,
         scale = "row",
         show_rownames = TRUE,
         border = FALSE,
         annotation = meta[, c("response"), drop = FALSE],
         main = "Top 50 Up- and Down- regulated genes in date_of: 20180323 vs 20180228",
         fontsize = 20)
```


Top 50 Up- and Down- regulated genes in date_of: 20180323 vs 20180228
response
4
2
0
-2
-4
response
non-pCR
pCR
Z84488.2
URB2
ACOT8
WDR24
TELO2
PEX11B
BCL9
MID1IP1
CREB3L4
KCNK1
SLC50A1
DNAJC1
HIST1H3D
HIST2H2AA3
C7orf55-LUC7L2
AP000350.4
PRSS33
POU3F2
GRIN2B
CHI3L1
MT-ND3
ZNF334
MUC6
KRT4
ISY1-RAB43
S3188-2
S3404-1
S3474-3
S3644-1
S4172-1
S4237-1
S4261-1
S3816-1
S4047-1
S3372-1
S3563-3
S3688-2
S3732-1
S3713-1
S3822-1
S3291-3
S3220-1
S3723-1
S3154-2
S3190-4
S4089-1
S4136-1

Generate input files for GSEA

```
#  prepares an expression profile for GSEA
#  http://software.broadinstitute.org/cancer/software/gsea/wiki/index.php/Data_formats#Expression_Data_
#  for GSEA it is important to report all genes - genome wide
#  hopefully cpms are better than logcpms
counts <- counts[rowSums(counts)>0,]
result_file <-  paste0("tables/1day.4gsea.txt")

counts_gsea <- counts %>% as.data.frame() %>%
    rownames_to_column(var = "ensembl_gene_id") %>%
    left_join(gene_symbol, by = c("ensembl_gene_id" = "gene_id")) %>%
    dplyr::relocate(symbol)
#%>%
#    dplyr::relocate(ensembl_gene_id)

colnames(counts_gsea)[1:2] <- c("NAME", "DESCRIPTION")

d <-duplicated(counts_gsea$NAME)
o <-  order(rowSums(counts_gsea[,rownames(meta)]),decreasing = T)
counts_gsea <- counts_gsea[o, ]
counts_gsea <- counts_gsea[!d, ]


samples_yes <- meta %>% dplyr::filter(response == "Yes") %>% row.names()
samples_no <- meta %>% dplyr::filter(response == "No") %>% row.names()

counts_gsea <- counts_gsea[,c("NAME", "DESCRIPTION", samples_yes, samples_no)]
# gsea now supports ENSEMBL_IDs
write_tsv(counts_gsea, result_file)
```

Functional analysis

Biological Process (BP)

```
bg_genes <- rownames(resResponse)

## Run GO enrichment analysis
compGO <- enrichGO(gene = sigResponse_up,
                   universe = bg_genes,
                   keyType = "ENSEMBL",
                   OrgDb = "org.Hs.eg.db",
                   ont = "BP",
                   qvalueCutoff  = 0.05,
                   pAdjustMethod = "BH",
                   readable = TRUE)
```

```
## Error in enrichGO(gene = sigResponse_up, universe = bg_genes, keyType = "ENSEMBL", : could not find
```

```
#dotplot(compGO,
#        showCategory = 20,
#        title = "GO (Biological Process) Enrichment \n Analysis for UP in Responders)",
#        label_format = 20,
#        font.size = 10)
# image pdf 12 x 12

## Output results from GO analysis to a table
print("UP")
```

```
## [1] "UP"
```

```
results_up <- data.frame(compGO@result) %>% dplyr::filter(p.adjust < 0.05)
```

```
## Error in data.frame(compGO@result): object 'compGO' not found
```

```
nrow(results_up)
```

```
## Error in h(simpleError(msg, call)): error in evaluating the argument 'x' in selecting a method for fu
```

```
write_csv(results_up, "tables/T21.day8.GO_BP_UP.csv")
```

```
## Error in is.data.frame(x): object 'results_up' not found
```

```
compGO <- enrichGO(gene = sigResponse_down,
                   universe = bg_genes,
                   keyType = "ENSEMBL",
                   OrgDb = "org.Hs.eg.db",
                   ont = "BP",
                   qvalueCutoff  = 0.05,
                   pAdjustMethod = "BH",
                   readable = TRUE)
```

```
## Error in enrichGO(gene = sigResponse_down, universe = bg_genes, keyType = "ENSEMBL", : could not fin
```

```
results_down <- data.frame(compGO@result) %>% dplyr::filter(p.adjust < 0.05)
```

```
## Error in data.frame(compGO@result): object 'compGO' not found
```

```
print("Down")
```

```
## [1] "Down"
```

```
nrow(results_down)
```

```
## Error in h(simpleError(msg, call)): error in evaluating the argument 'x' in selecting a method for fu
```

R session

```
sessionInfo()
```

```
## R version 4.0.3 (2020-10-10)
## Platform: x86_64-redhat-linux-gnu (64-bit)
## Running under: Fedora 32 (Workstation Edition)
##
## Matrix products: default
## BLAS/LAPACK: /usr/lib64/libopenblas-r0.3.12.so
##
## locale:
##  [1] LC_CTYPE=en_CA.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_CA.UTF-8        LC_COLLATE=en_CA.UTF-8
##  [5] LC_MONETARY=en_CA.UTF-8    LC_MESSAGES=en_CA.UTF-8
##  [7] LC_PAPER=en_CA.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_CA.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] ensembldb_2.14.1            AnnotationFilter_1.14.0
##  [3] GenomicFeatures_1.42.3      AnnotationDbi_1.52.0
##  [5] AnnotationHub_2.22.1        BiocFileCache_1.14.0
##  [7] dbplyr_2.1.1                knitr_1.30
##  [9] ggrepel_0.9.1               tximport_1.18.0
## [11] DEGreport_1.26.0            pheatmap_1.0.12
## [13] RColorBrewer_1.1-2          forcats_0.5.1
## [15] stringr_1.4.0               dplyr_1.0.5
## [17] purrr_0.3.4                 readr_1.4.0
## [19] tidyr_1.1.3                 tibble_3.1.1
## [21] ggplot2_3.3.3               tidyverse_1.3.1
## [23] DESeq2_1.30.1               SummarizedExperiment_1.20.0
## [25] Biobase_2.50.0              MatrixGenerics_1.2.1
## [27] matrixStats_0.58.0          GenomicRanges_1.42.0
## [29] GenomeInfoDb_1.26.7         IRanges_2.24.1
## [31] S4Vectors_0.28.1            BiocGenerics_0.36.1
##
## loaded via a namespace (and not attached):
##   [1] readxl_1.3.1                backports_1.2.1
##   [3] circlize_0.4.12             plyr_1.8.6
##   [5] lazyeval_0.2.2              ConsensusClusterPlus_1.54.0
##   [7] splines_4.0.3               BiocParallel_1.24.1
##   [9] digest_0.6.27               htmltools_0.5.1.1
##  [11] fansi_0.4.2                 magrittr_2.0.1
##  [13] memoise_2.0.0               cluster_2.1.0
##  [15] limma_3.46.0                ComplexHeatmap_2.6.2
##  [17] Biostrings_2.58.0           annotate_1.68.0
##  [19] Nozzle.R1_1.1-1             modelr_0.1.8
##  [21] askpass_1.1                 prettyunits_1.1.1
##  [23] colorspace_2.0-0            blob_1.2.1
##  [25] rvest_1.0.0                 rappdirs_0.3.3
##  [27] haven_2.4.1                 xfun_0.19
##  [29] crayon_1.4.1                RCurl_1.98-1.3
##  [31] jsonlite_1.7.2              genefilter_1.72.1
##  [33] survival_3.2-7              glue_1.4.2
##  [35] gtable_0.3.0                zlibbioc_1.36.0
##  [37] XVector_0.30.0              MatrixModels_0.5-0
##  [39] GetoptLong_1.0.5            DelayedArray_0.16.3
##  [41] shape_1.4.5                 SparseM_1.81
##  [43] scales_1.1.1                DBI_1.1.1
##  [45] edgeR_3.32.1                Rcpp_1.0.6
##  [47] progress_1.2.2              xtable_1.8-4
##  [49] lasso2_1.2-21.1             tmvnsim_1.0-2
##  [51] clue_0.3-59                 bit_4.0.4
##  [53] httr_1.4.2                  ellipsis_0.3.1
##  [55] farver_2.1.0                  pkgconfig_2.0.3
##  [57] reshape_0.8.8                 XML_3.99-0.6
##  [59] locfit_1.5-9.4                utf8_1.2.1
##  [61] labeling_0.4.2                tidyselect_1.1.0
##  [63] rlang_0.4.10                  later_1.2.0
##  [65] munsell_0.5.0                 BiocVersion_3.12.0
##  [67] cellranger_1.1.0              tools_4.0.3
##  [69] cachem_1.0.4                  cli_2.5.0
##  [71] generics_0.1.0                RSQLite_2.2.7
##  [73] broom_0.7.6                   evaluate_0.14
##  [75] fastmap_1.1.0                 ggdendro_0.1.22
##  [77] yaml_2.2.1                    bit64_4.0.5
##  [79] fs_1.5.0                      nlme_3.1-149
##  [81] quantreg_5.85                 mime_0.9
##  [83] xml2_1.3.2                    biomaRt_2.46.3
##  [85] compiler_4.0.3                rstudioapi_0.13
##  [87] curl_4.3                      png_0.1-7
##  [89] interactiveDisplayBase_1.28.0 reprex_2.0.0
##  [91] geneplotter_1.68.0            stringi_1.5.3
##  [93] lattice_0.20-41               ProtGenerics_1.22.0
##  [95] Matrix_1.2-18                 psych_2.1.3
##  [97] vctrs_0.3.7                   pillar_1.6.0
##  [99] lifecycle_1.0.0               BiocManager_1.30.12
## [101] GlobalOptions_0.1.2           conquer_1.0.2
## [103] cowplot_1.1.1                 bitops_1.0-7
## [105] rtracklayer_1.50.0            httpuv_1.6.0
## [107] R6_2.5.0                      promises_1.2.0.1
## [109] MASS_7.3-53                   assertthat_0.2.1
## [111] openssl_1.4.3                 rjson_0.2.20
## [113] withr_2.4.2                   GenomicAlignments_1.26.0
## [115] Rsamtools_2.6.0               mnormt_2.0.2
## [117] GenomeInfoDbData_1.2.4        hms_1.0.0
## [119] grid_4.0.3                    rmarkdown_2.5
## [121] Cairo_1.5-12.2                logging_0.10-108
## [123] shiny_1.6.0                   lubridate_1.7.10
```