

# DE analysis

Sergey Naumenko

2021-05-08

## Contents

<b>Overview</b>	<b>2</b>
<b>Checking to see that the transcript to gene mapping is correct</b>	<b>2</b>
<b>Sanity check that metadata matches your expression</b>	<b>2</b>
<b>Run DESeq2</b>	<b>3</b>
<b>Wald test</b>	<b>3</b>
<b>DEGreport QC</b>	<b>4</b>
Size factor QC - samples 1-15 . . . . .	4
Size factor QC - samples 16-30 . . . . .	5
Size factor QC - samples 31-40 (44) . . . . .	6
<b>Mean-Variance QC plots</b>	<b>7</b>
treatment . . . . .	7
response . . . . .	8
ER . . . . .	9
tumor_percentage_high . . . . .	10
<b>Covariates effect on count data</b>	<b>11</b>
<b>Covariates correlation with metrics</b>	<b>13</b>
<b>Sample-level QC analysis</b>	<b>18</b>
PCA - treatment . . . . .	18
PCA - response . . . . .	19
PCA - ER . . . . .	20
PCA - date_of . . . . .	22
<b>Inter-correlation analysis</b>	<b>24</b>
Without study_id . . . . .	24
Without study_id = top 1000 variable genes . . . . .	26
Without study_id = top 500 variable genes . . . . .	28
With study_id . . . . .	30
<b>Treatment Post vs Pre - see Table3</b>	<b>32</b>
<b>Response pCR vs non-pCR - see Table4</b>	<b>32</b>
<b>ER : Positive vs Negative - Table 5</b>	<b>32</b>

tumor_percentage_high : High vs Low - Table 6	32
date_of: 20180323 vs 20180228 - Table 7	32
Visualization	33
Heatmaps	39
R session	49

## Overview

- Principal Investigator: Beth Overmoyer
- Experiment: RNAseq\_analysis\_of\_inflammatory\_breast\_cancer\_hbc04141
- study 6 was excluded because of 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>

## 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    44
```

## 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

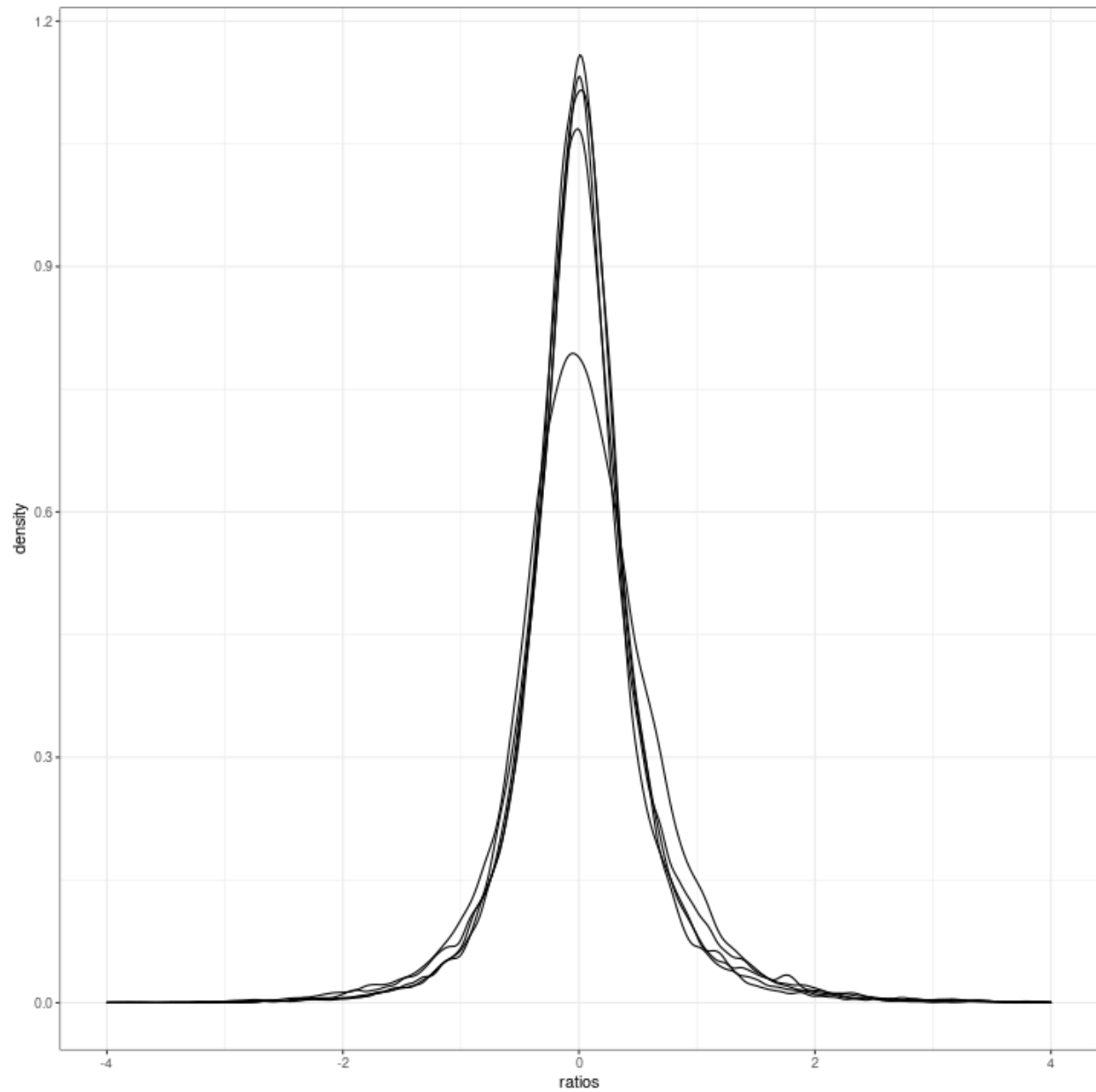
## Wald test

*Here we subset protein coding genes.*

# DEGreport QC

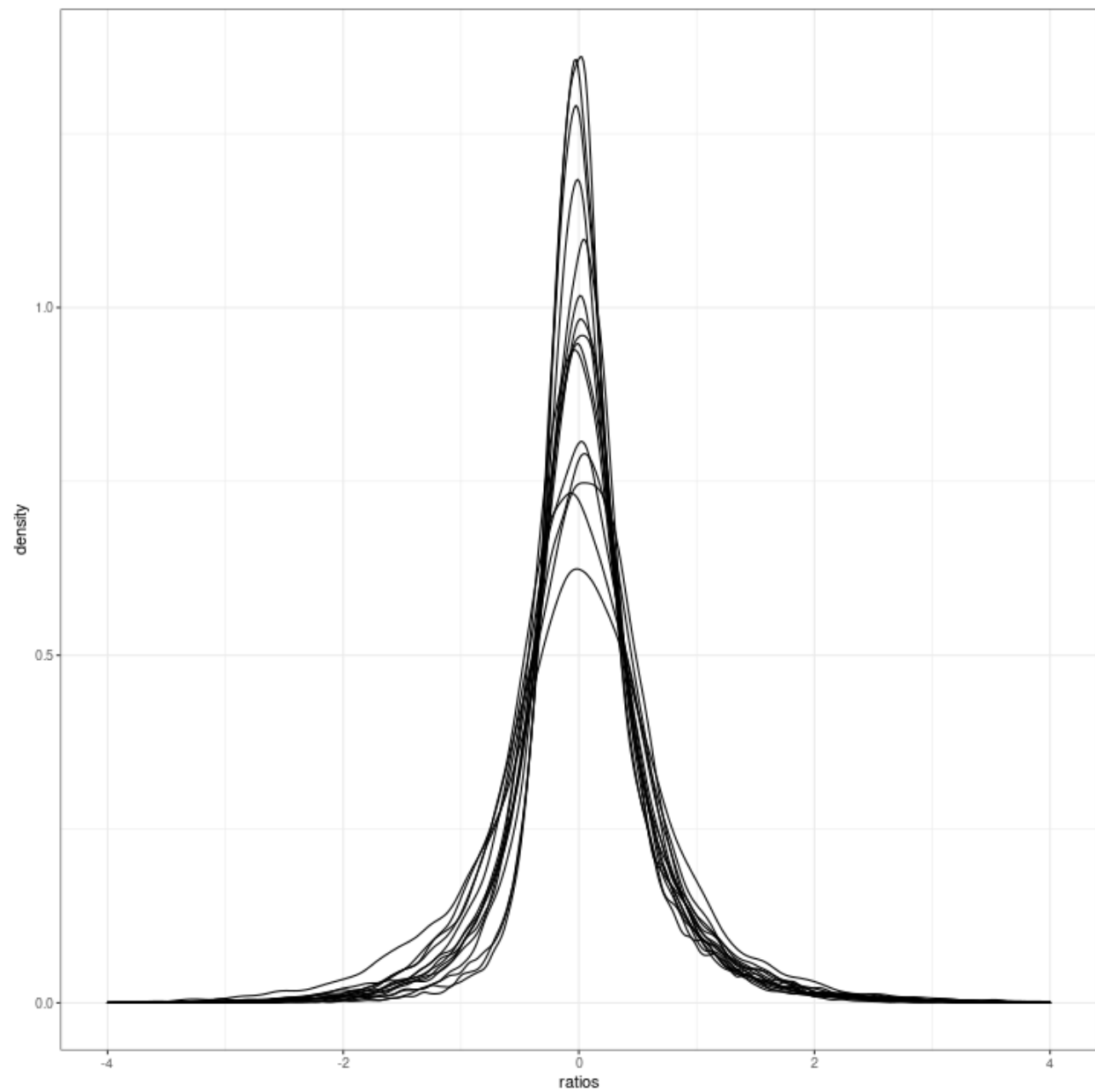
## Size factor QC - samples 1-15

```
counts <- counts(dds, normalized = TRUE)
design <- as.data.frame(colData(dds))
degCheckFactors(counts[, 1:5])
```



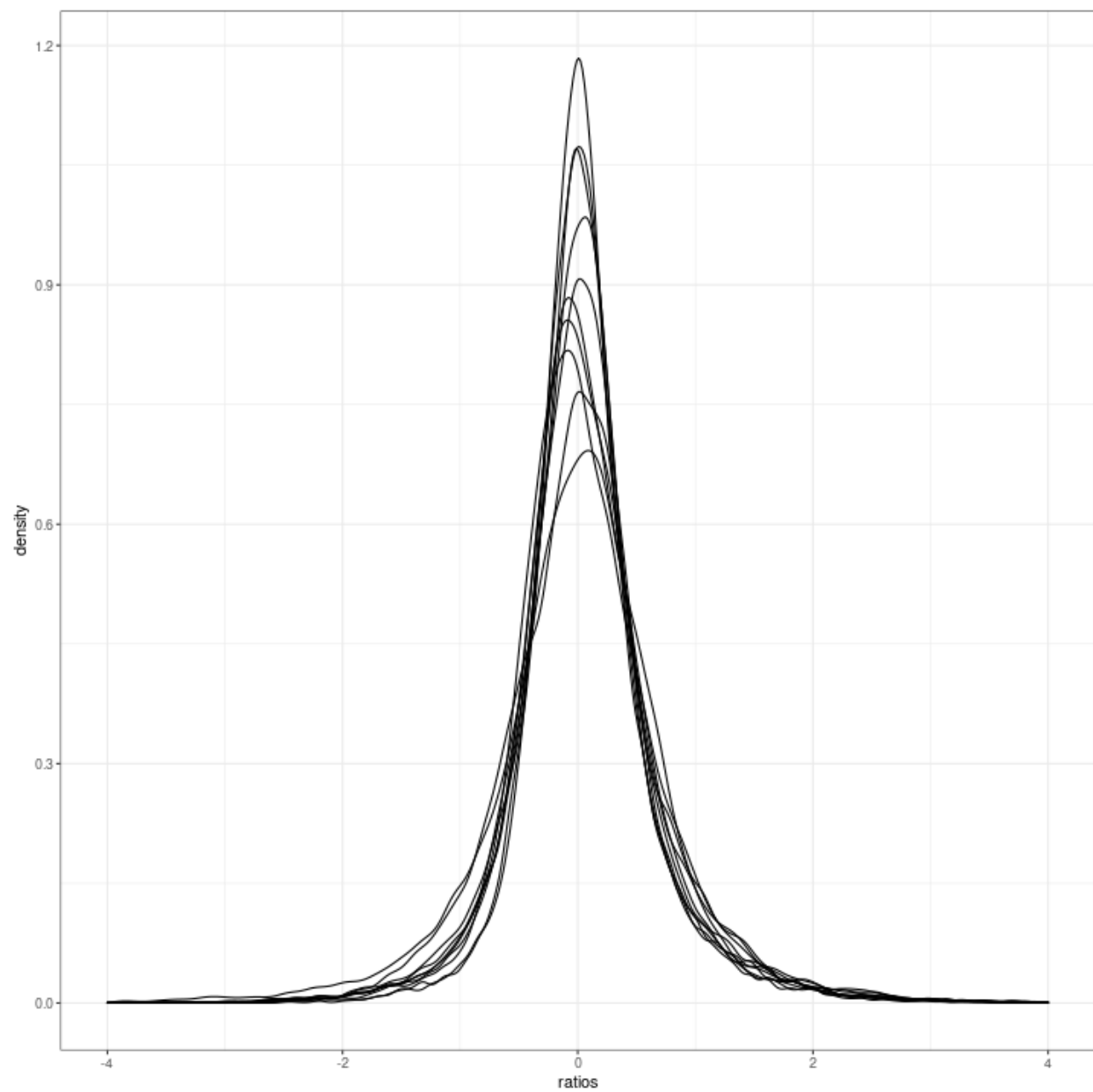
## Size factor QC - samples 16-30

```
degCheckFactors(counts[, 16:30])
```



## Size factor QC - samples 31-40 (44)

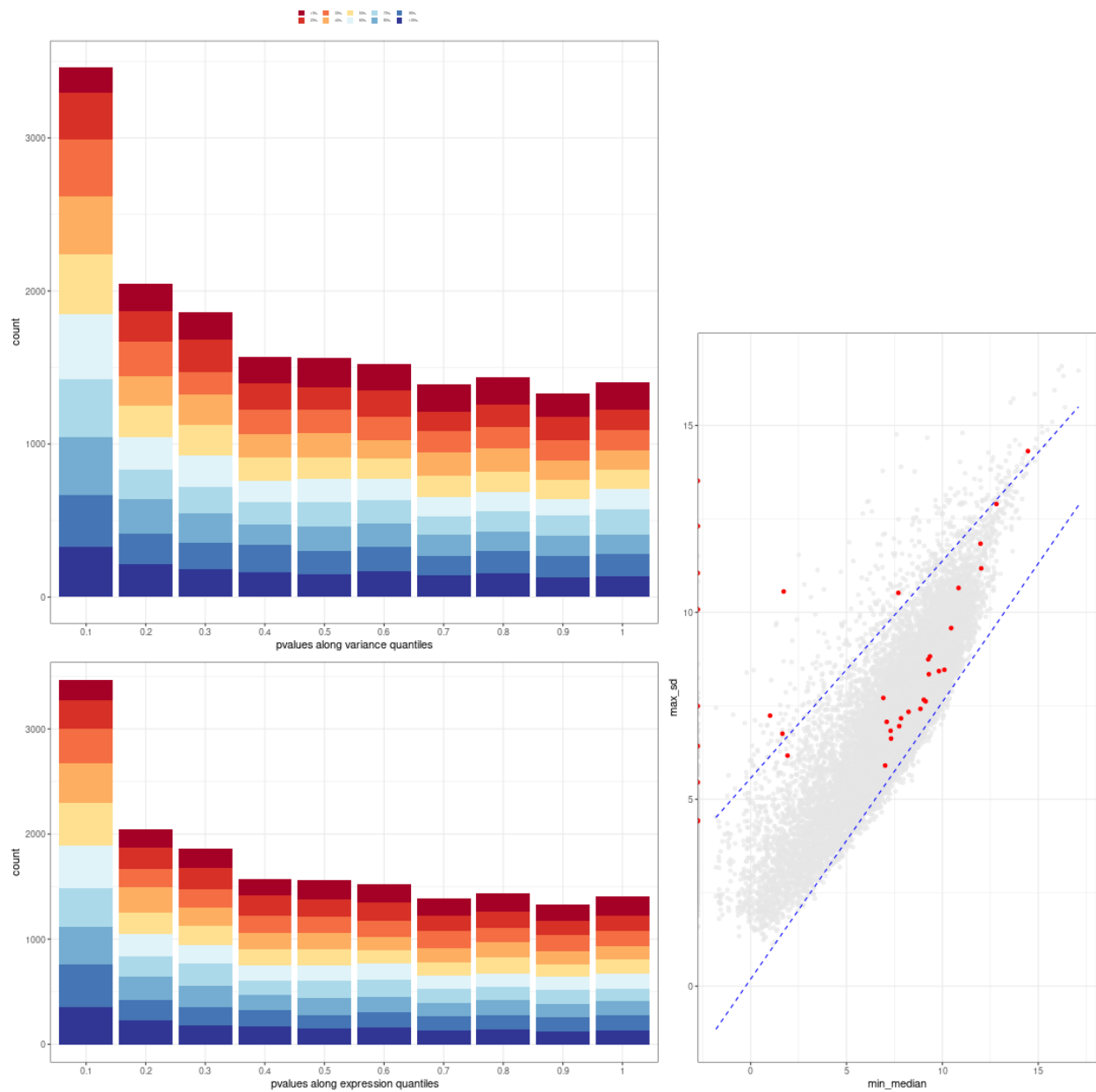
```
degCheckFactors(counts[, 31:ncol(counts)])
```



# Mean-Variance QC plots

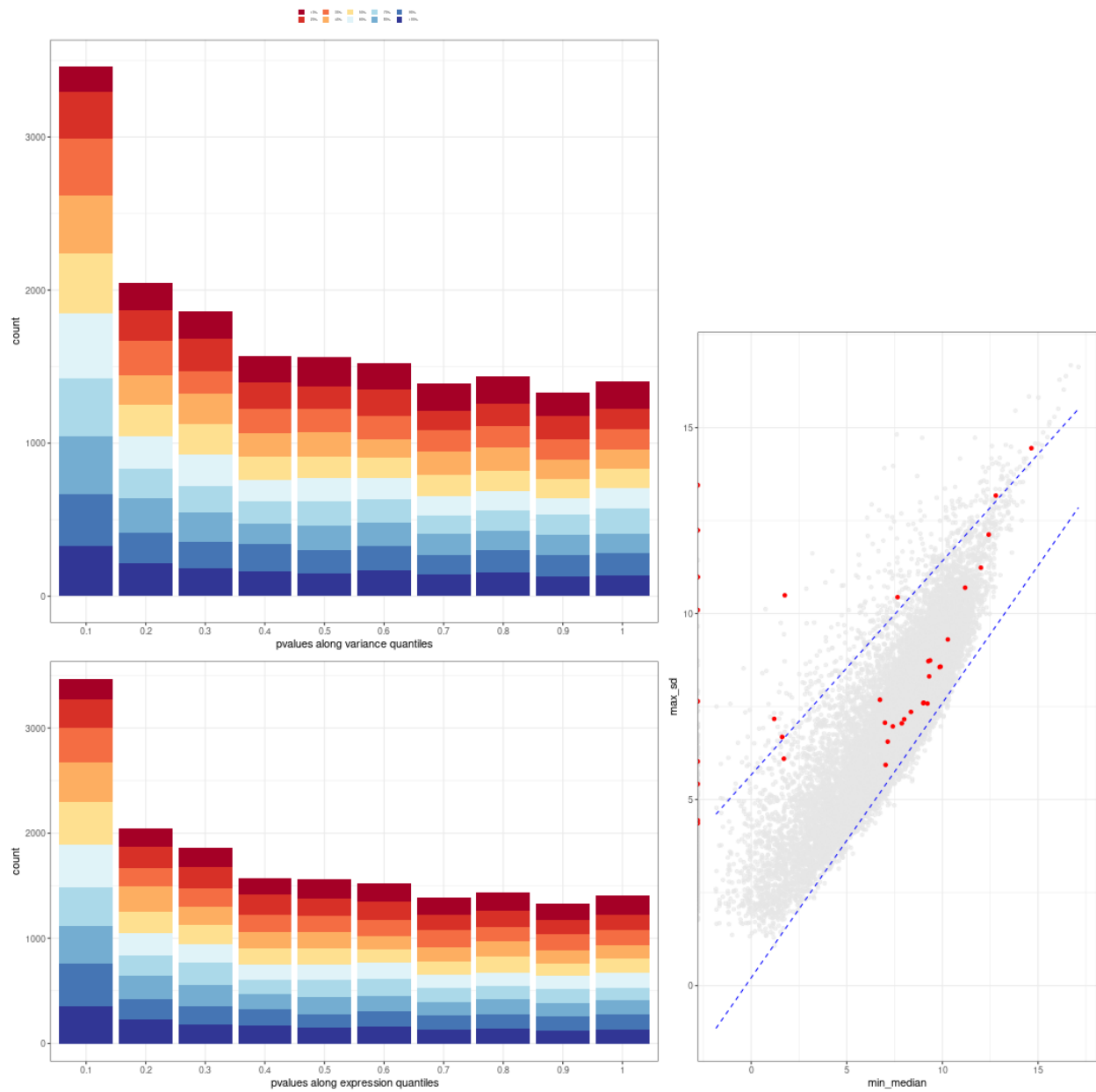
treatment

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



## response

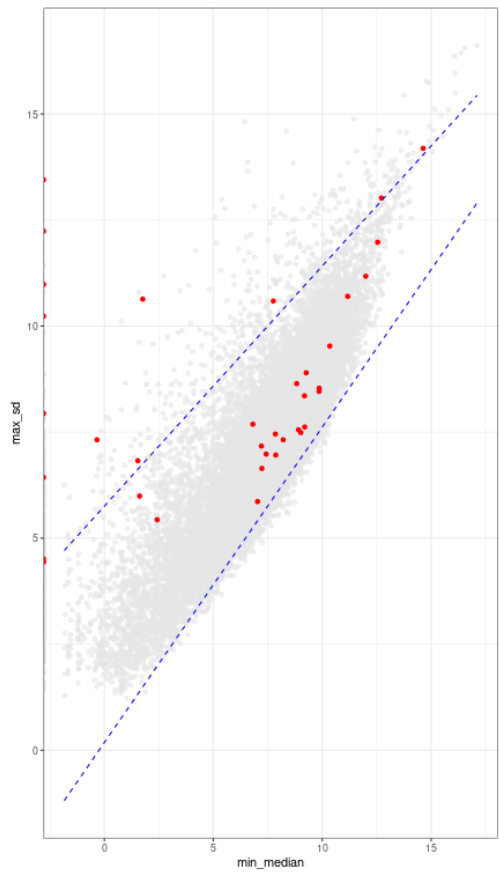
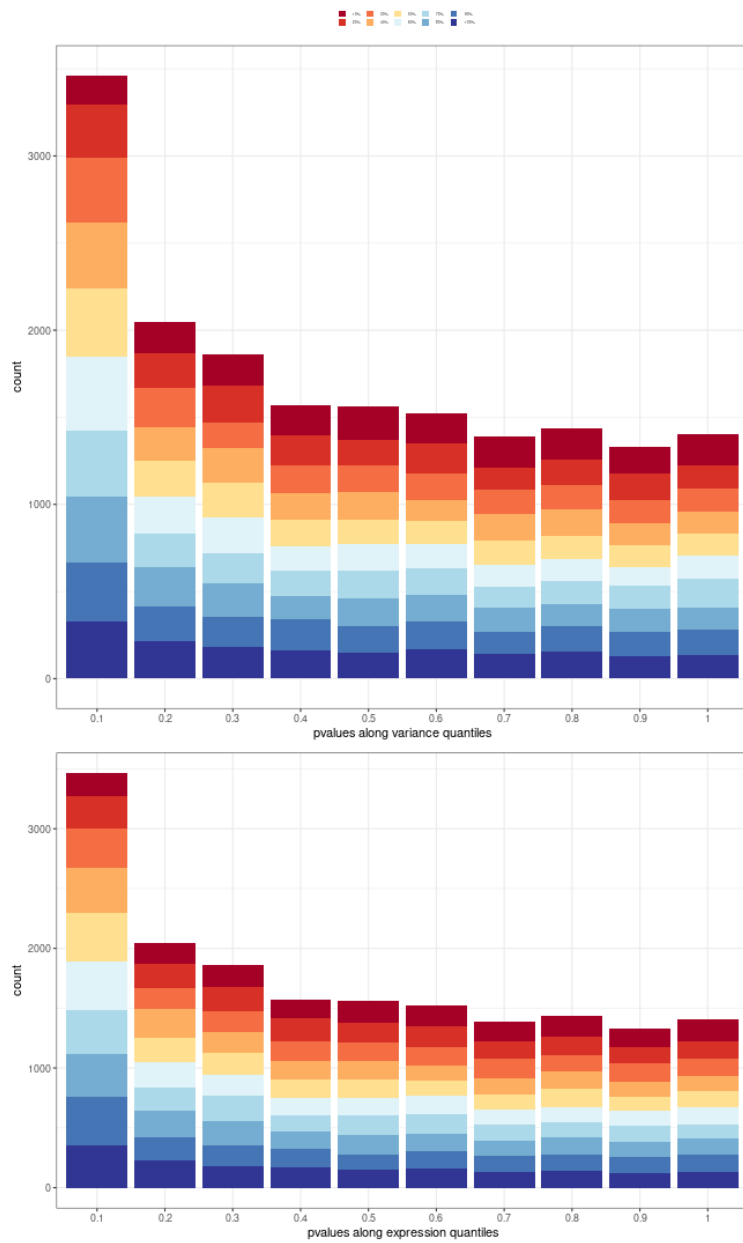
```
degQC(counts, design[["response"]], pvalue = res[["pvalue"]])
```





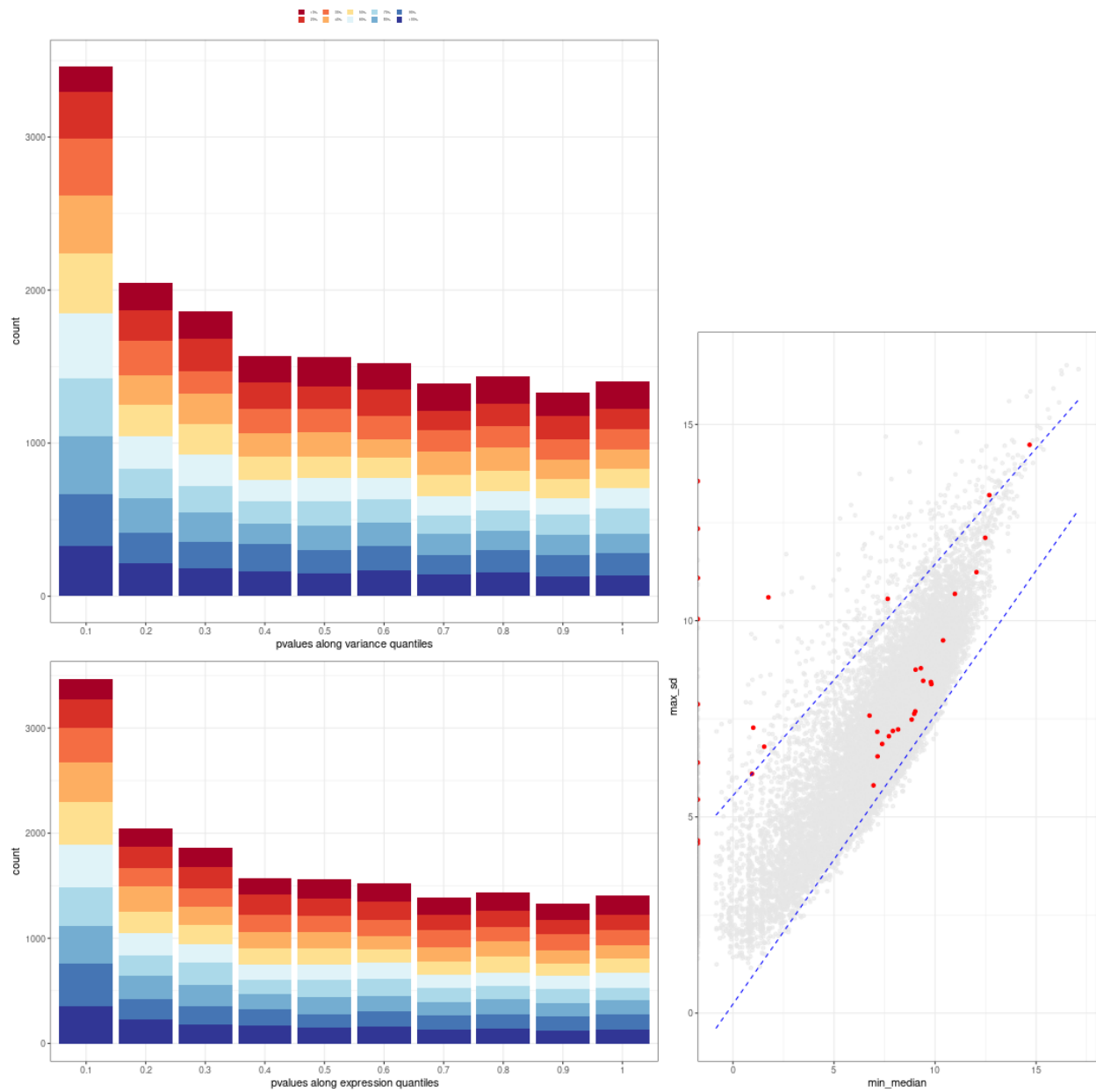
ER

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



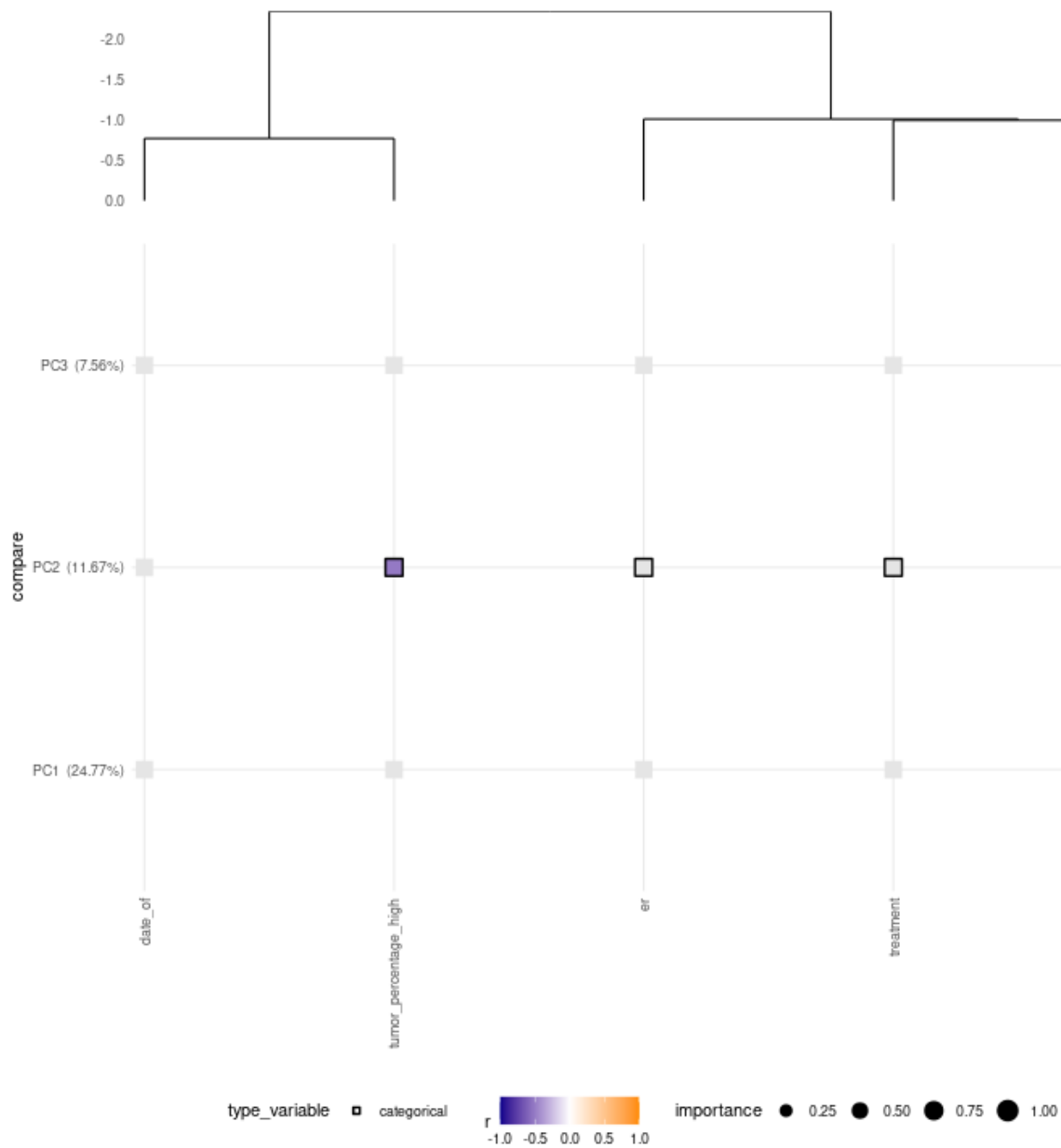
## tumor\_percentage\_high

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

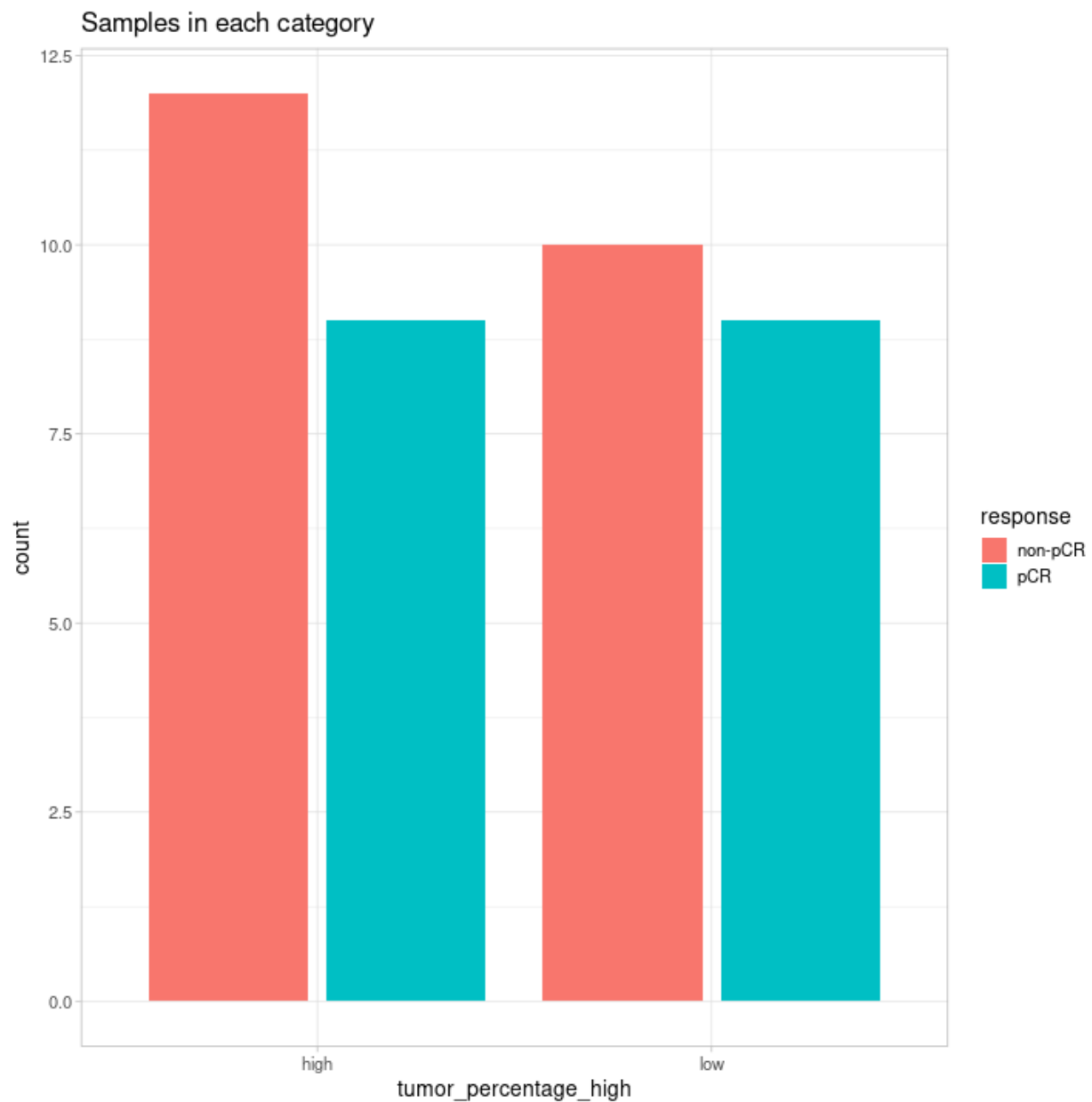


## Covariates effect on count data

```
mdata <- colData(dds) %>% as.data.frame() %>%  
  dplyr::select(treatment, response, er, date_of, tumor_percentage_high)  
resCov <- degCovariates(log2(counts(dds)+0.5), mdata)
```



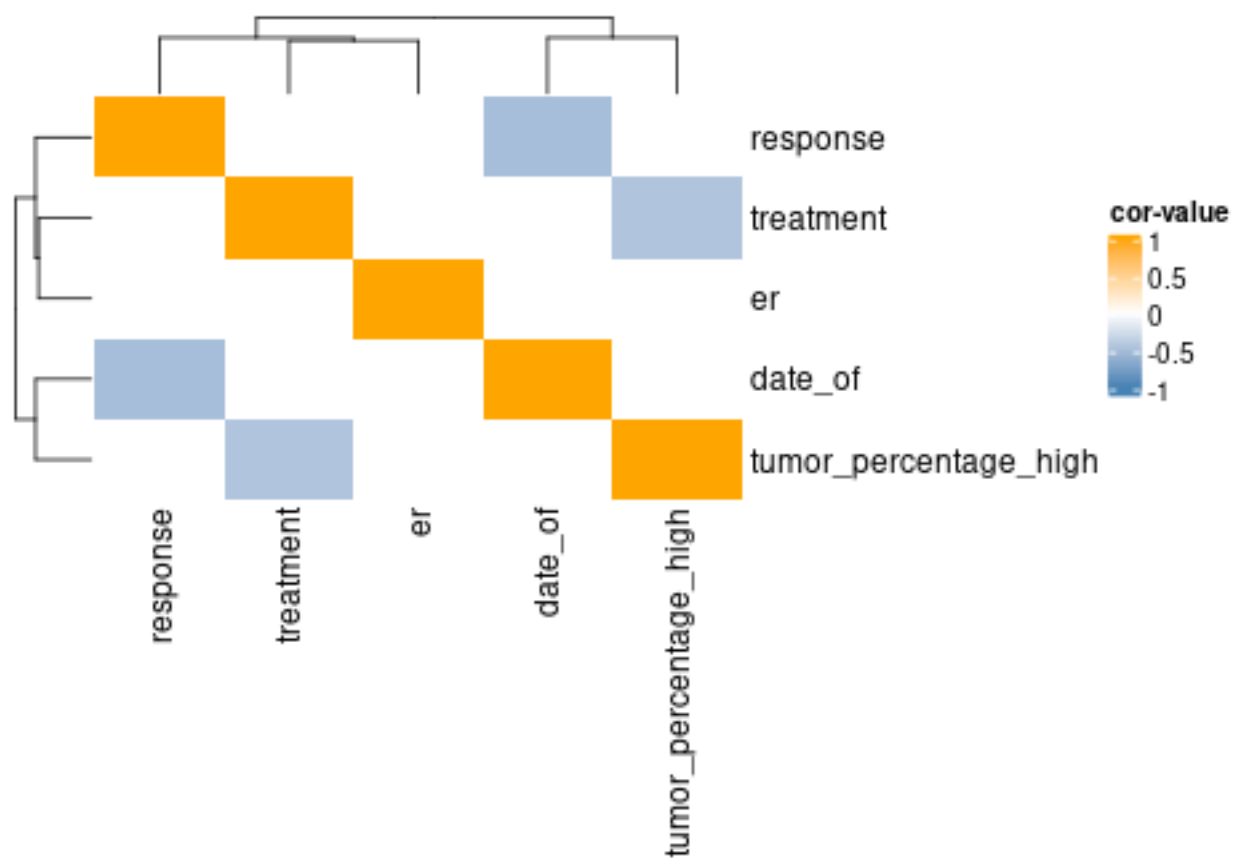
```
mdata %>% ggplot(aes(tumor_percentage_high, fill = response)) +  
  geom_bar(position = "dodge2") +  
  ggtitle("Samples in each category")
```



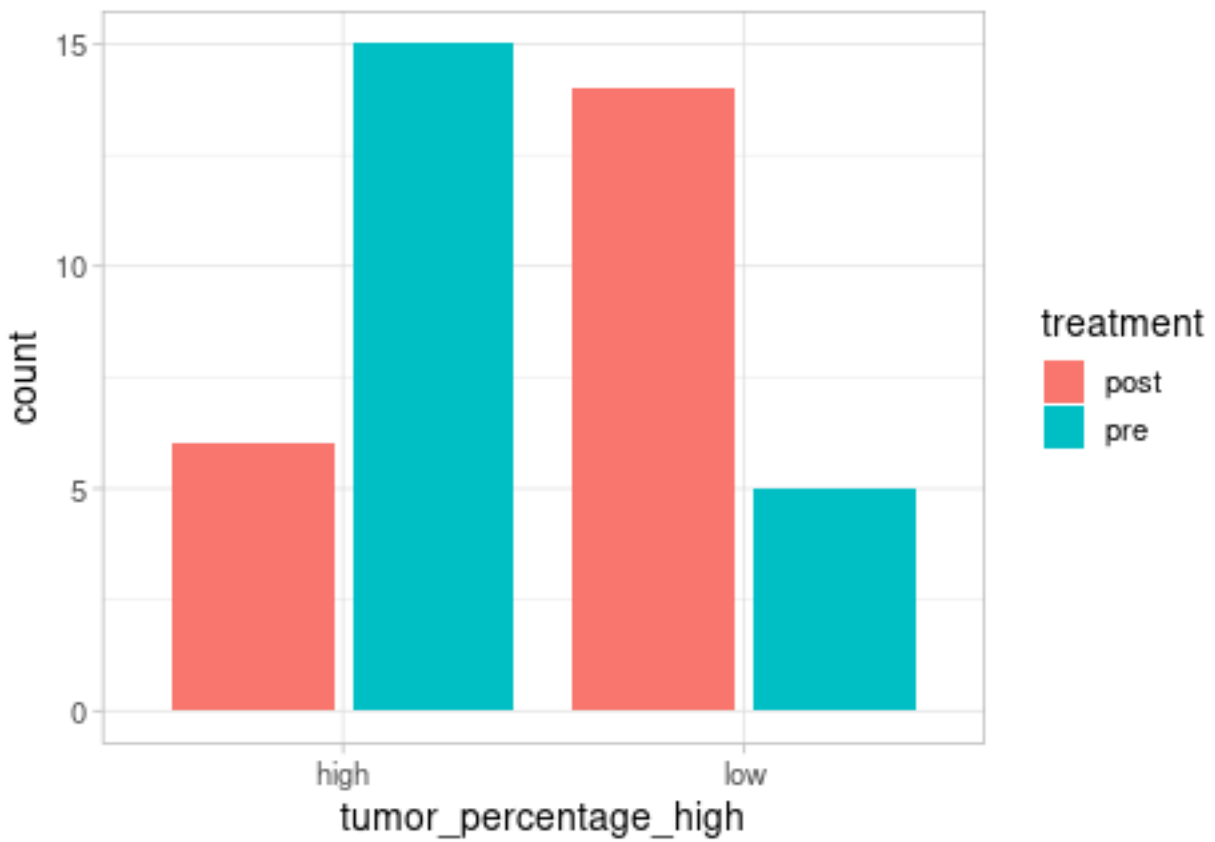
> Samples split equally between tumor\_percentage\_high for both response types. That allows to control for tumor\_percentage\_high batch effectively.

## Covariates correlation with metrics

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



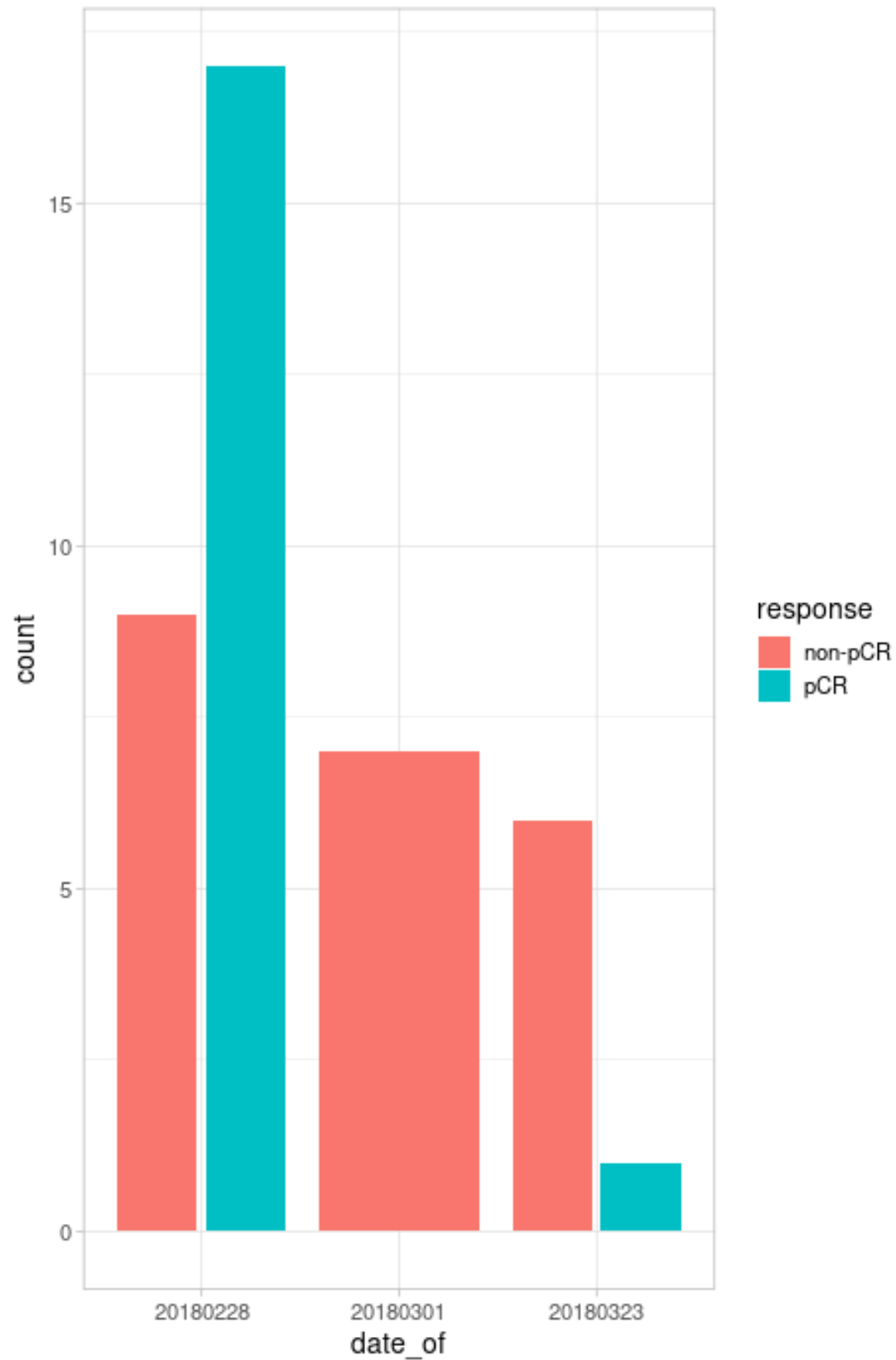
```
mdata %>% ggplot(aes(tumor_percentage_high, fill = treatment)) + geom_bar(position = "dodge2")
```



> pre-treatment samples have a larger proportion of higher tumor\_percentage - tumor content decreases after treatment, it is harder to sample high purity tumors.

```
mdata %>% ggplot(aes(date_of, fill = response)) +  
  geom_bar(position = "dodge2") +  
  ggtitle("Distribution of samples across dates of sequencing")
```

Distribution of samples across dates of sequencing



> Most

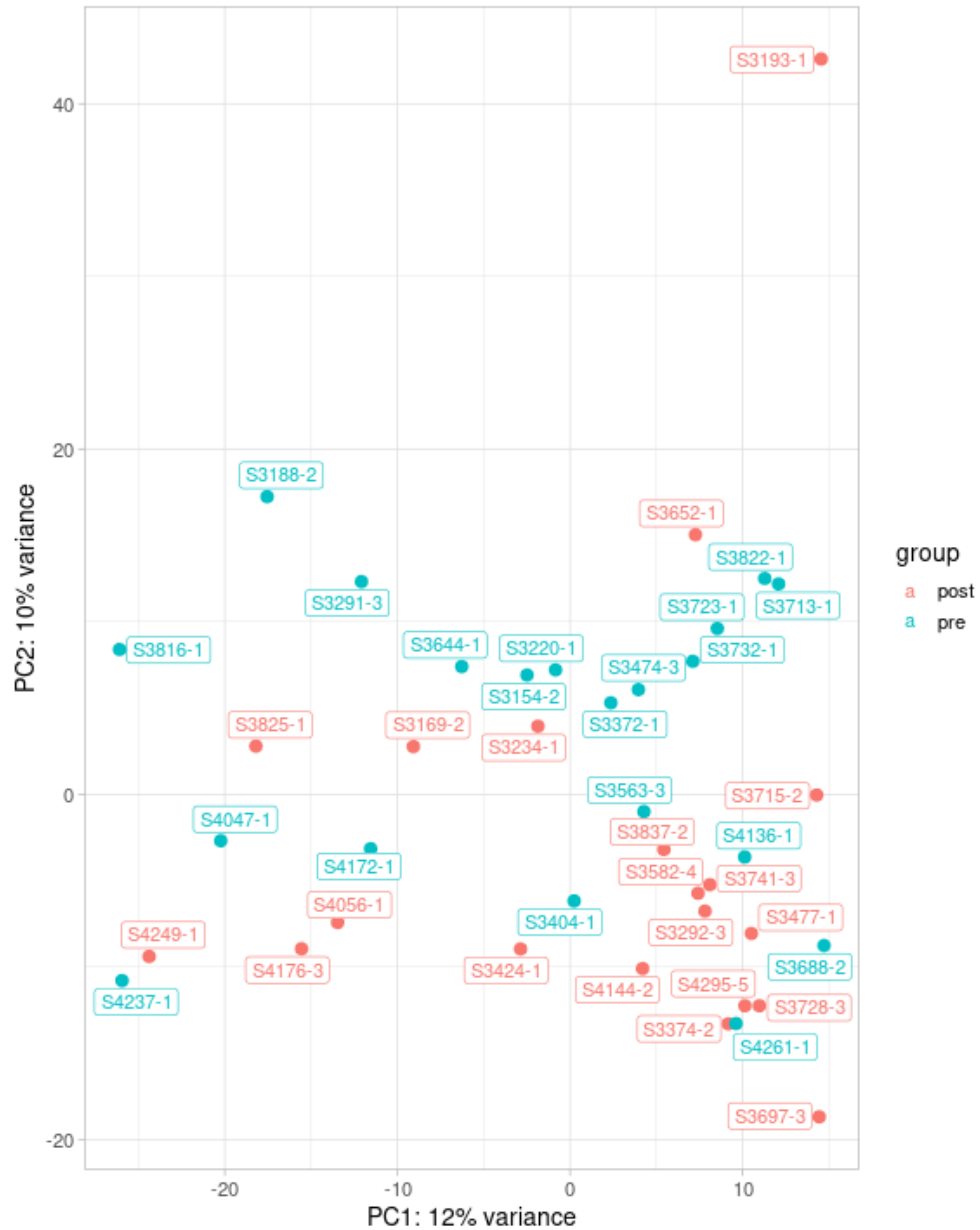


pCR samples were sequenced on 20180228 and non-pCR on two other dates. It is important to check the magnitude of DE signal between dates.

## Sample-level QC analysis

### PCA - treatment

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

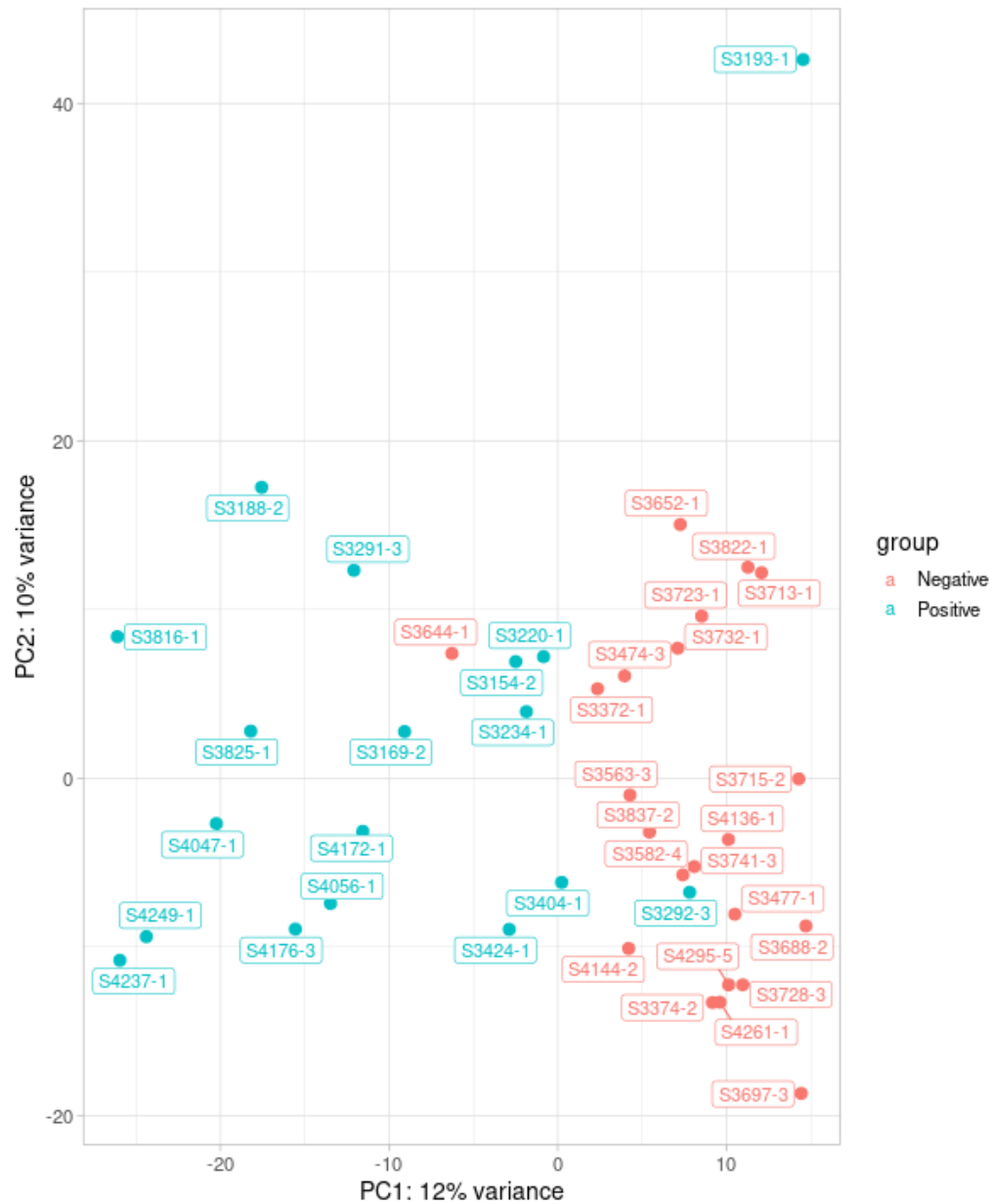


```
# Use the DESeq2 function
plotPCA(rld, intgroup = c("response")) + geom_label_repel(aes(label = name))
```



## PCA - ER

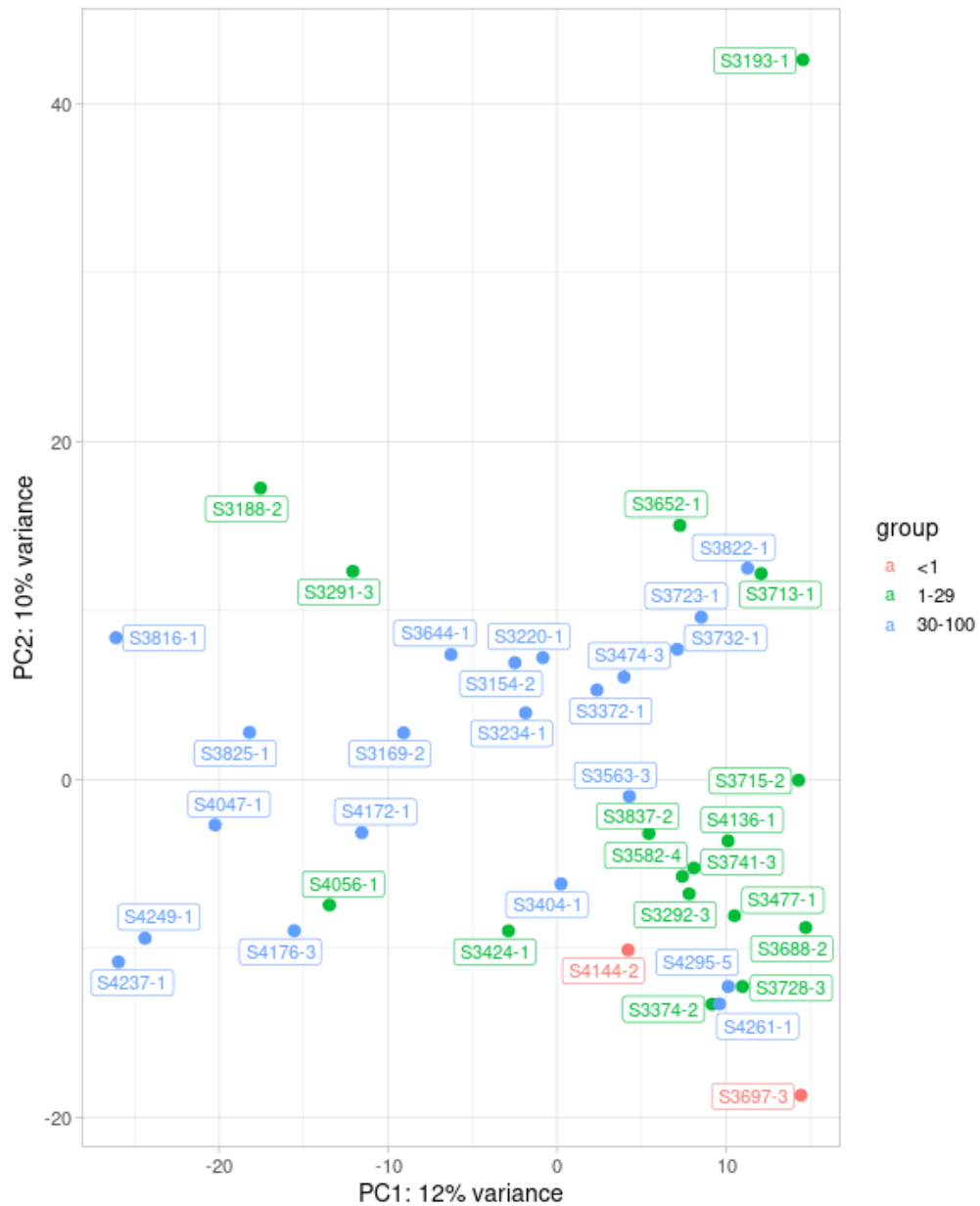
```
# Use the DESeq2 function
plotPCA(rld, intgroup = c("er")) + geom_label_repel(aes(label = name))
```



```
## PCA - tumor_percentage
```

```
# Use the DESeq2 function
```

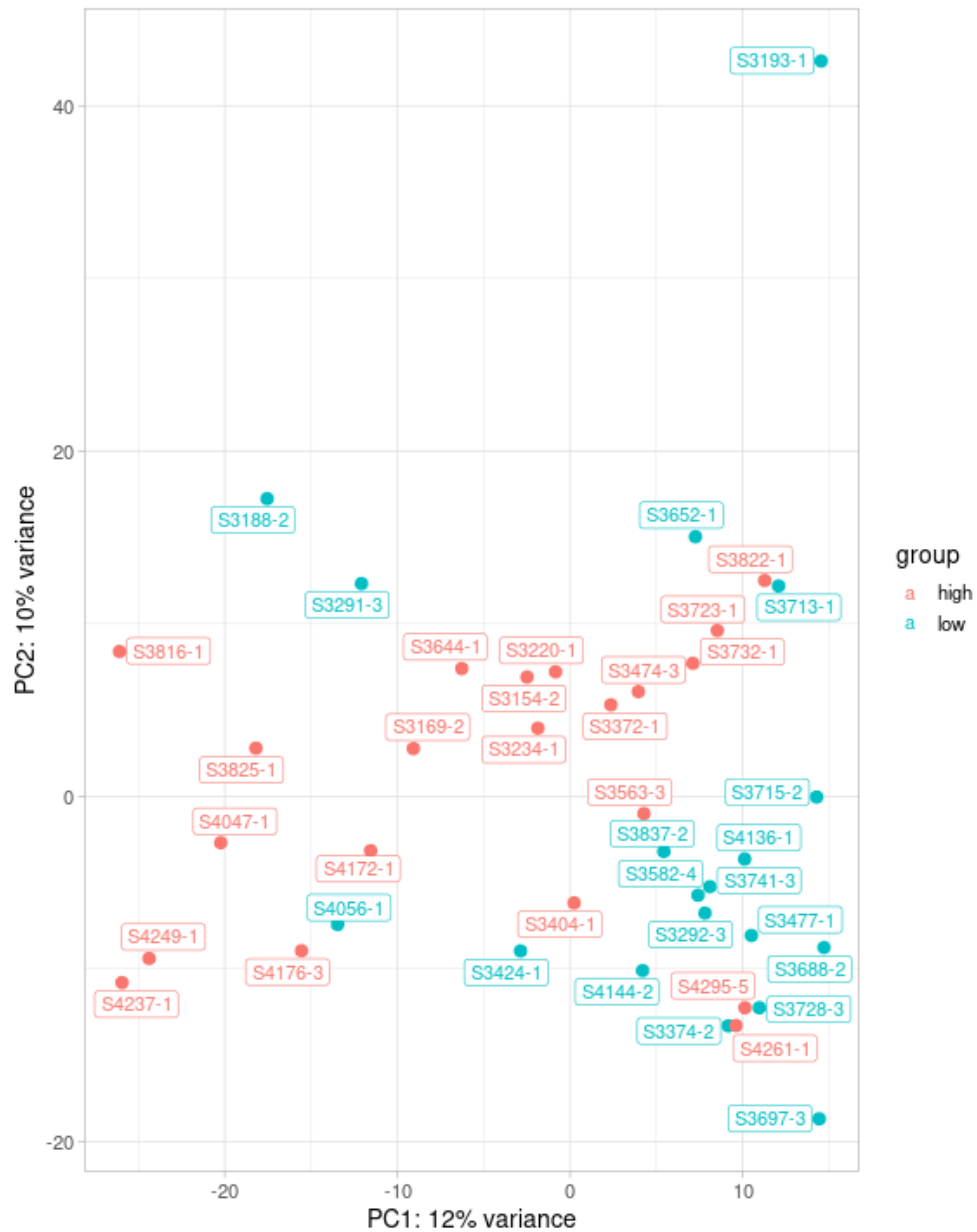
```
plotPCA(rld, intgroup = c("tumor_percentage")) + geom_label_repel(aes(label = name))
```



```
## PCA - tumor_percentage_high
```

```
# Use the DESeq2 function
```

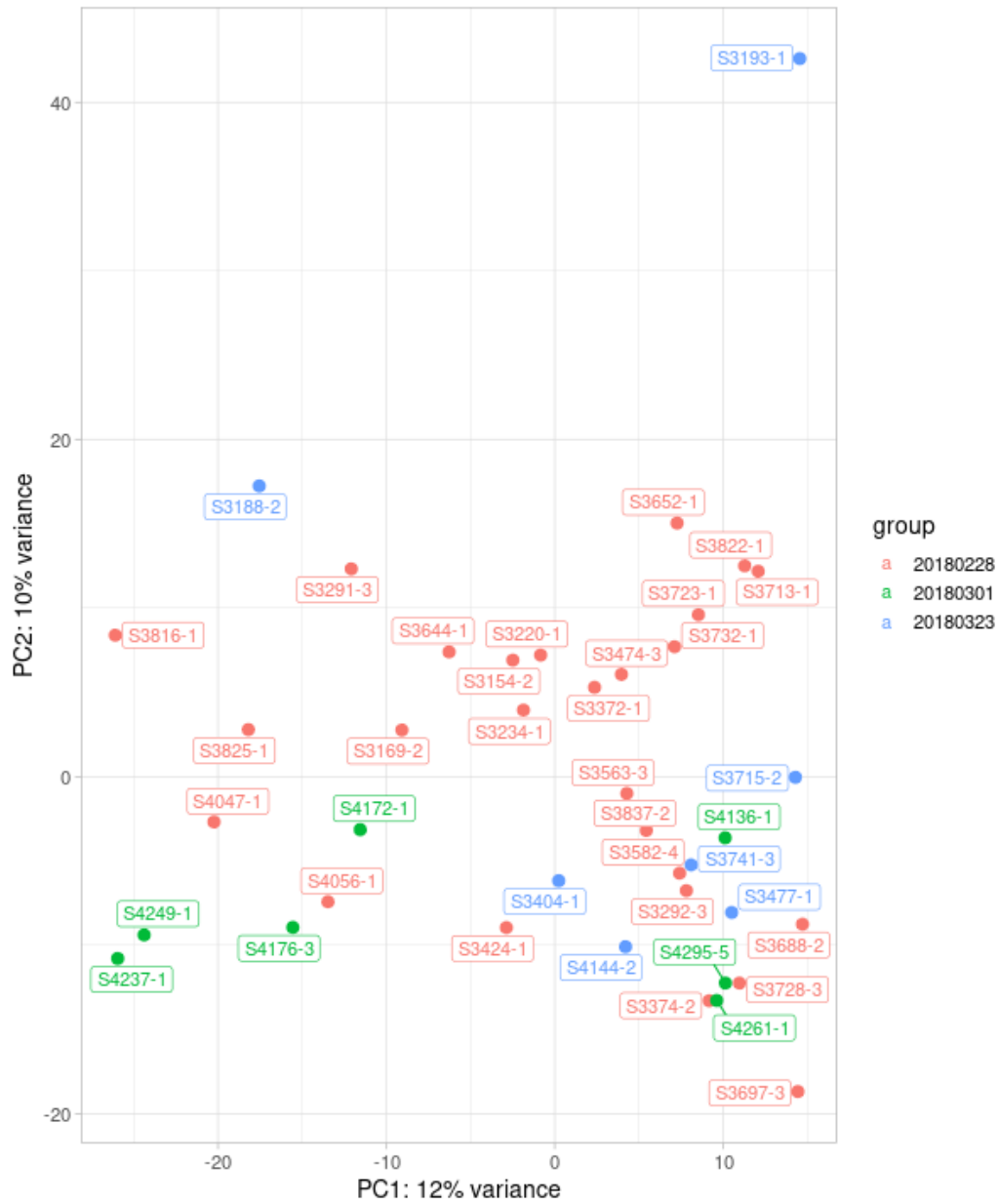
```
plotPCA(rld, intgroup = c("tumor_percentage_high")) + geom_label_repel(aes(label = name))
```



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

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

meta$study_id <- as.factor(meta$study_id)
# Create annotation file for samples
annotation <- meta[, c("treatment", "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)
```





Without study\_id = top 1000 variable genes

```
rv <- rowVars(rld_mat)
rv <- order(rv, decreasing = TRUE) %>% head(1000)
rld_mat_1000 <- rld_mat[rv,]
annotation <- meta[, c("treatment", "response", "er", "tumor_percentage_high", "date_of")]

# Change colors
heat.colors <- brewer.pal(6, "Blues")
rld_cor <- cor(rld_mat_1000)
# Plot heatmap
pheatmap(rld_cor,
          annotation = annotation,
          border = NA,
          fontsize = 20)
```



Without study\_id = top 500 variable genes

```
rv <- rowVars(rld_mat)
rv <- order(rv, decreasing = TRUE) %>% head(500)
rld_mat_500 <- rld_mat[rv,]
annotation <- meta[, c("treatment", "response", "er", "tumor_percentage_high", "date_of")]

# Change colors
heat.colors <- brewer.pal(6, "Blues")
rld_cor <- cor(rld_mat_500)
# Plot heatmap
pheatmap(rld_cor,
          annotation = annotation,
          border = NA,
          fontsize = 20)
```



## With study\_id

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

meta$study_id <- as.factor(meta$study_id)
# Create annotation file for samples
annotation <- meta[, c("treatment", "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)
```



Treatment Post vs Pre - see Table3

Response pCR vs non-pCR - see Table4

ER : Positive vs Negative - Table 5

tumor\_\_percentage\_\_high : High vs Low - Table 6

date\_\_of: 20180323 vs 20180228 - Table 7

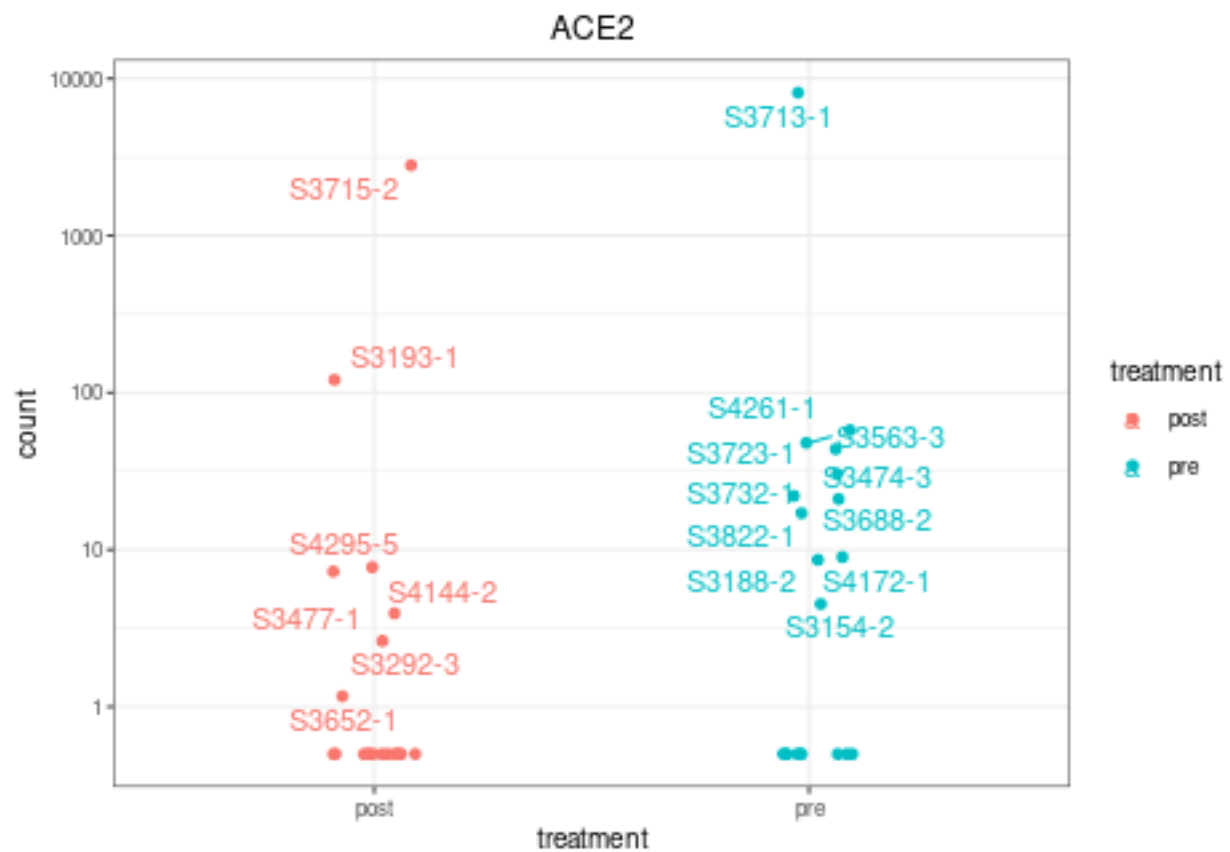


## Visualization

Gene example

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

ggplot(d, aes(x = treatment, y = count, color = treatment)) +
  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()
```

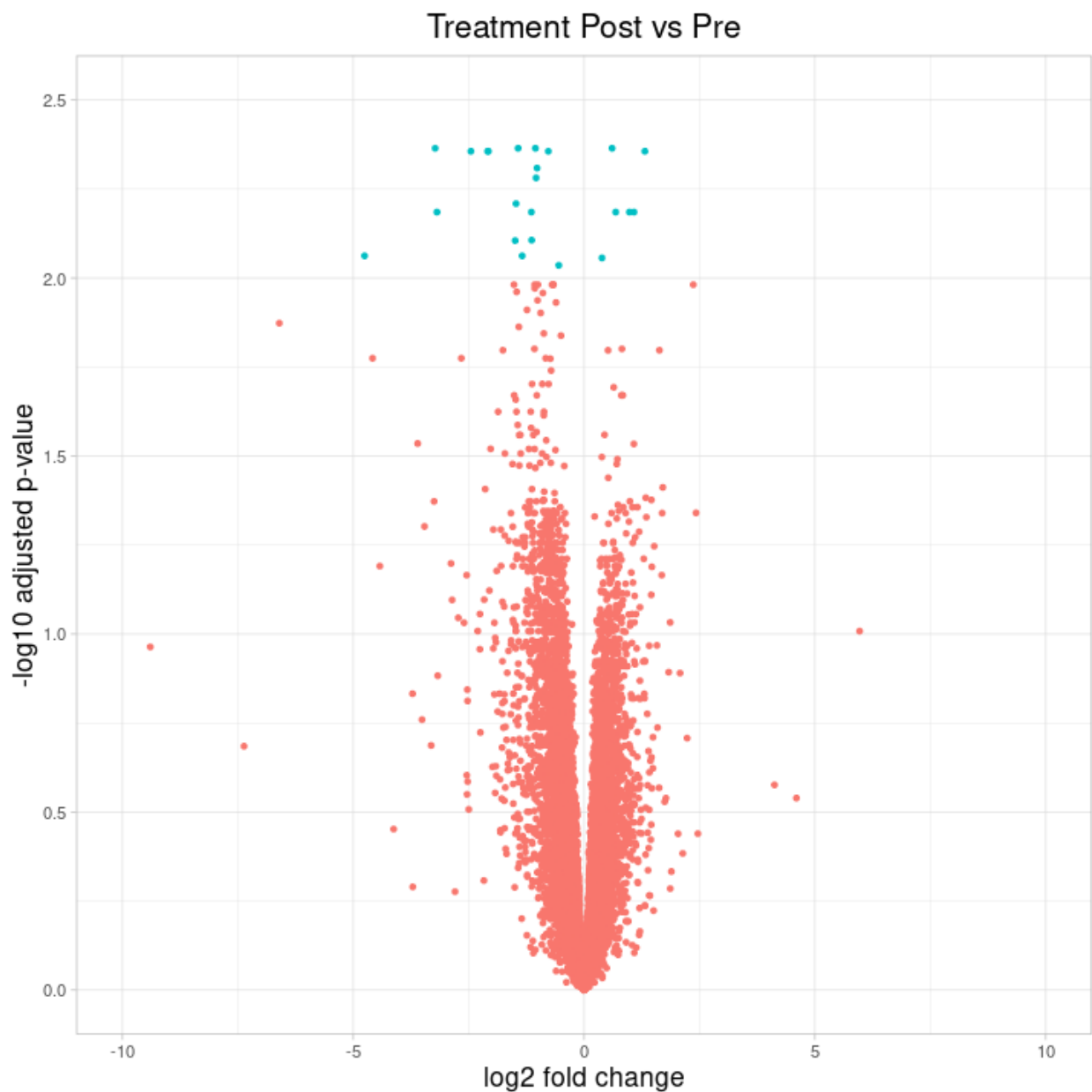


```

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

## Volcano plot
ggplot(resTreatment_tb) +
  geom_point(aes(x = log2FoldChange, y = -log10(padj), colour = threshold)) +
  ggtitle("Treatment Post vs Pre") +
  xlab("log2 fold change") +
  ylab("-log10 adjusted p-value") +
  scale_x_continuous(limits = c(-10,10)) +
  scale_y_continuous(limits = c(0, 2.5)) +
  theme(legend.position = "none",
        plot.title = element_text(size = rel(1.5), hjust = 0.5),
        axis.title = element_text(size = rel(1.25)))

```

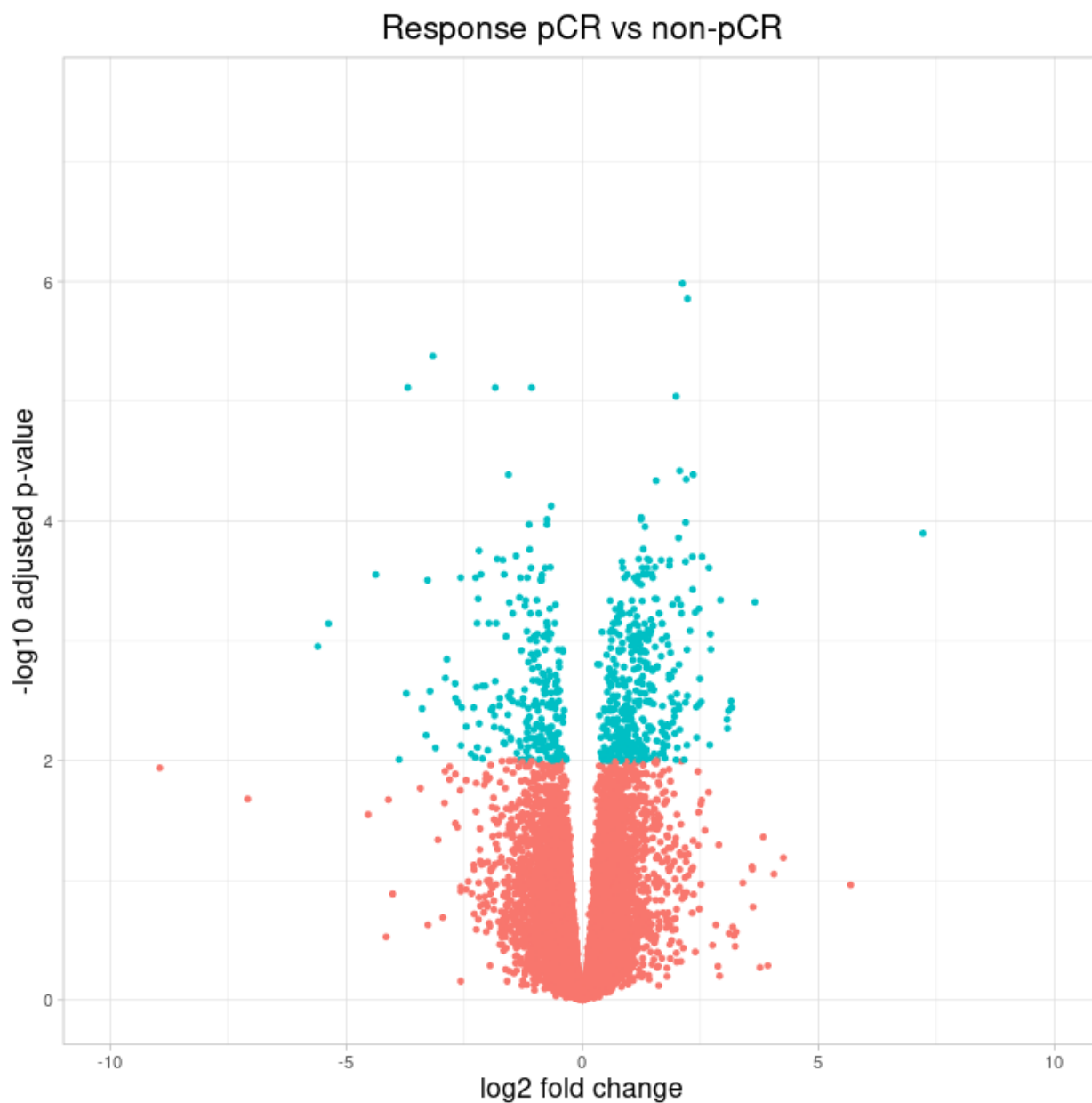


```

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

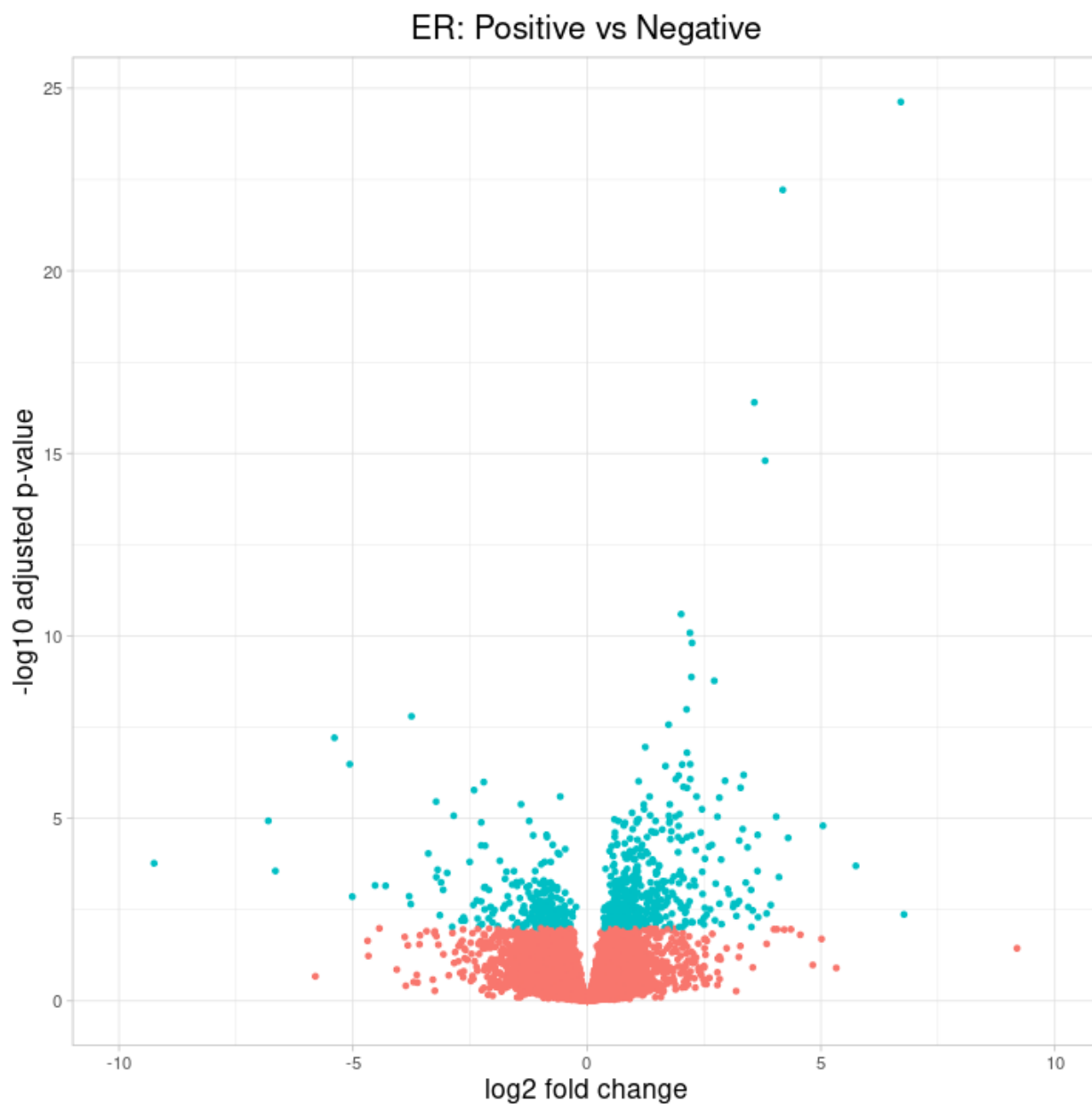
ggplot(resResponse_tb) +
  geom_point(aes(x = log2FoldChange, y = -log10(padj), colour = threshold)) +
  ggtitle("Response pCR vs non-pCR") +
  xlab("log2 fold change") +
  ylab("-log10 adjusted p-value") +
  scale_x_continuous(limits = c(-10,10)) +
  scale_y_continuous(limits = c(0, 7.5))+
  theme(legend.position = "none",
        plot.title = element_text(size = rel(1.5), hjust = 0.5),
        axis.title = element_text(size = rel(1.25)))

```



```
# 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)))
```

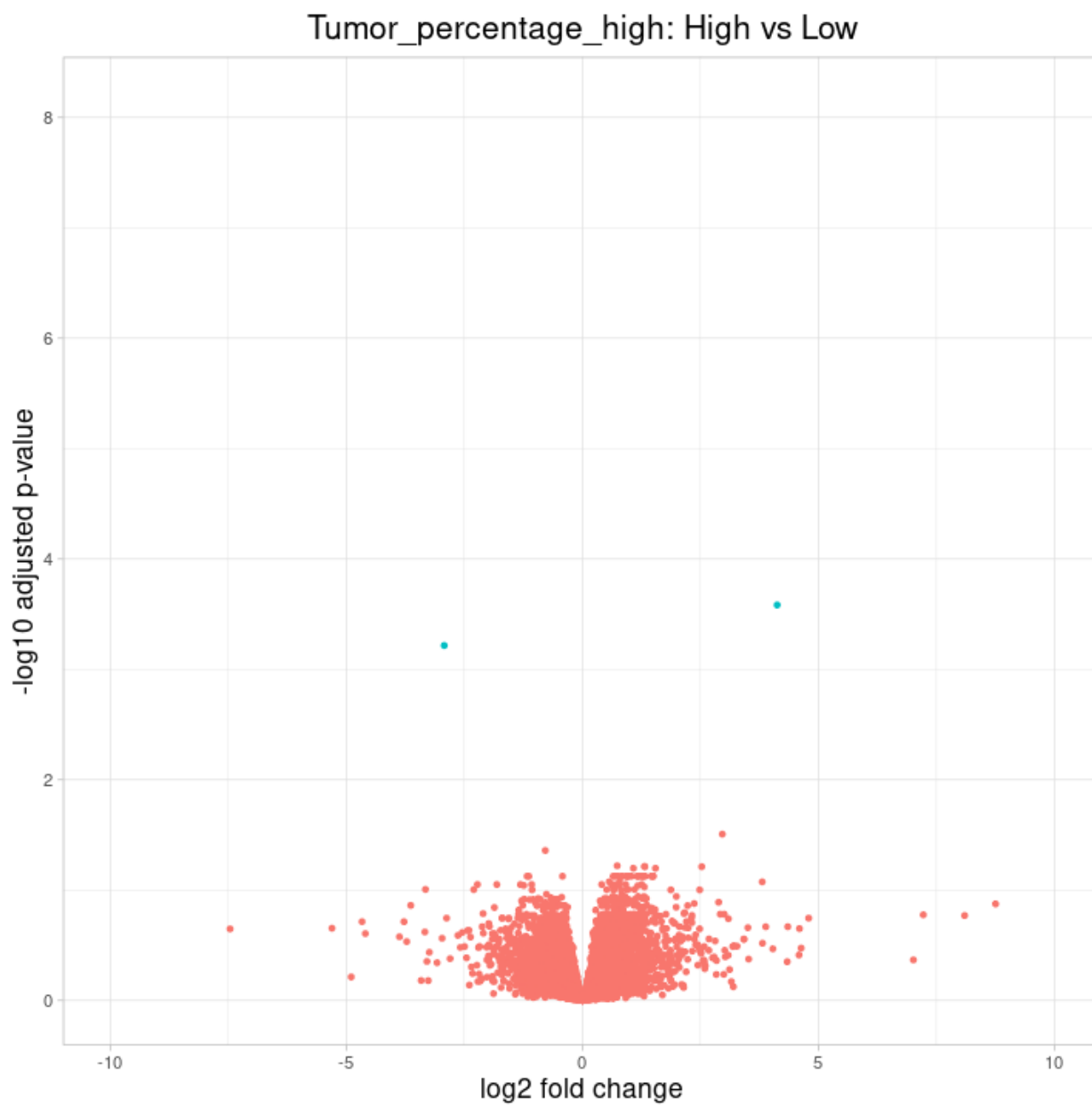


```

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

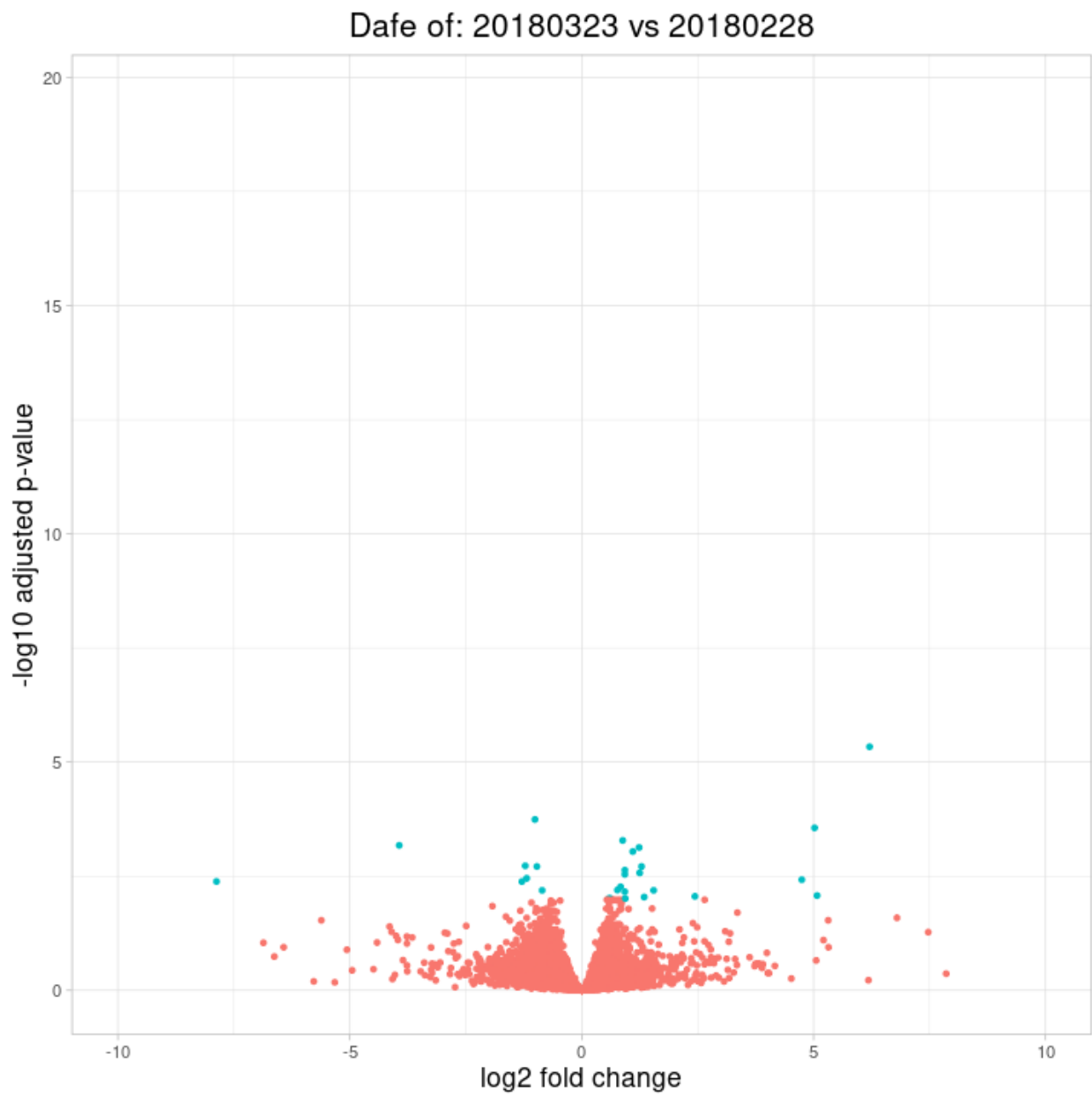
ggplot(resTP_tb) +
  geom_point(aes(x = log2FoldChange, y = -log10(padj), colour = threshold)) +
  ggtitle("Tumor_percentage_high: High vs Low") +
  xlab("log2 fold change") +
  ylab("-log10 adjusted p-value") +
  scale_x_continuous(limits = c(-10,10)) +
  theme(legend.position = "none",
        plot.title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

```



```
# Add a column for significant genes
resD0_tb <- resD0_tb %>% mutate(threshold = padj < 0.01)

ggplot(resD0_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 <- resTreatment_tb_significant %>% arrange(-log2FoldChange) %>% head(50) %>% pull(gene)
sig_down <- resTreatment_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)

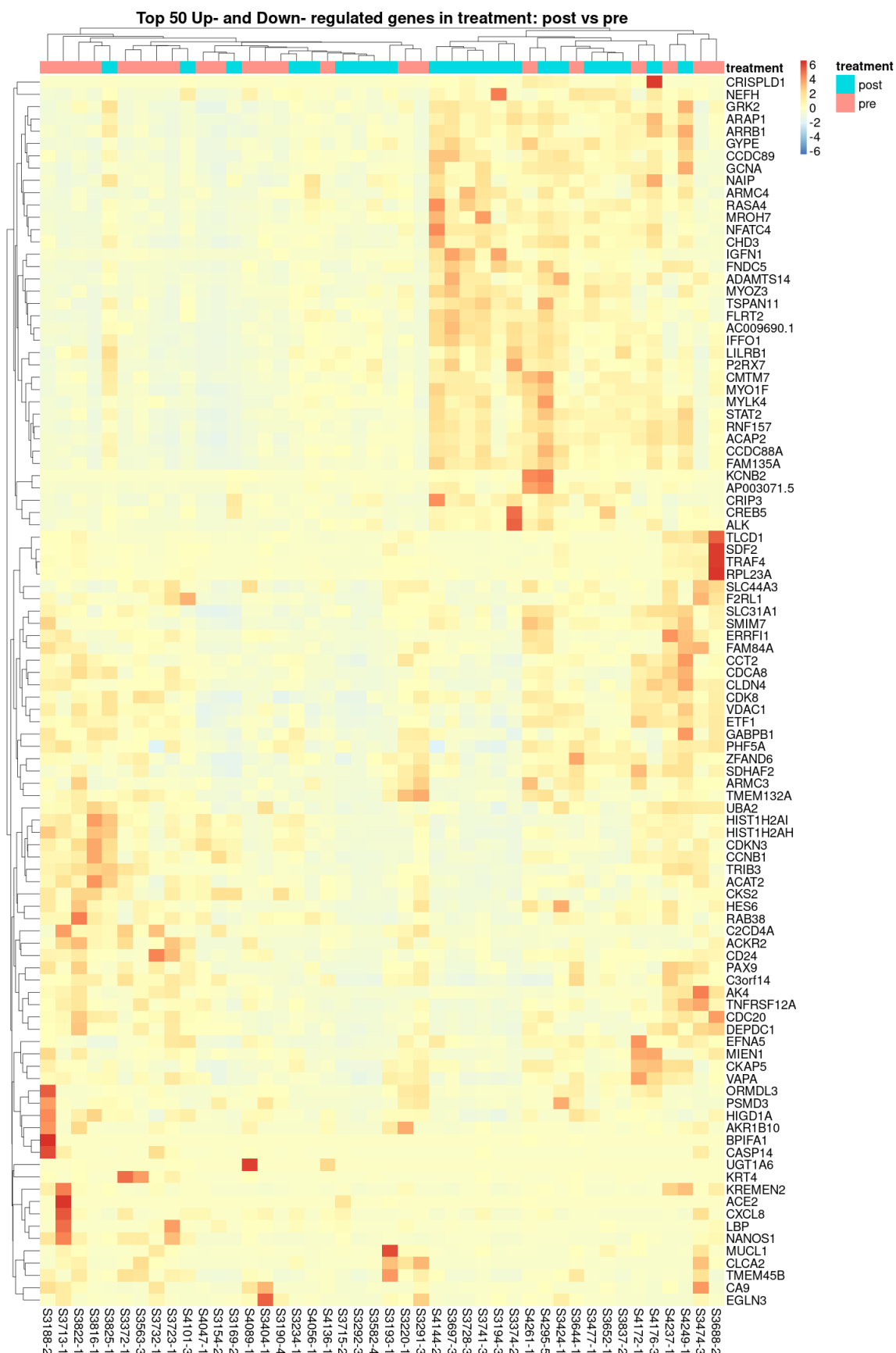
plotmat <- txi$abundance[c(sig_up, sig_down),] %>% as.data.frame() %>%
  rownames_to_column(var = "ensembl_gene_id") %>%
  left_join(gene_symbol, by = c("ensembl_gene_id" = "gene_id")) %>%
  drop_na(symbol)

plotmat$ensembl_gene_id <- NULL

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

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

# Plot heatmap
# color = heat.colors,
pheatmap(plotmat, scale = "row",
  show_rownames = TRUE,
  border = FALSE,
  annotation = meta[, c("treatment"), drop = FALSE],
  main = "Top 50 Up- and Down- regulated genes in treatment: post vs pre",
  fontsize = 20)
```





```

# 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)

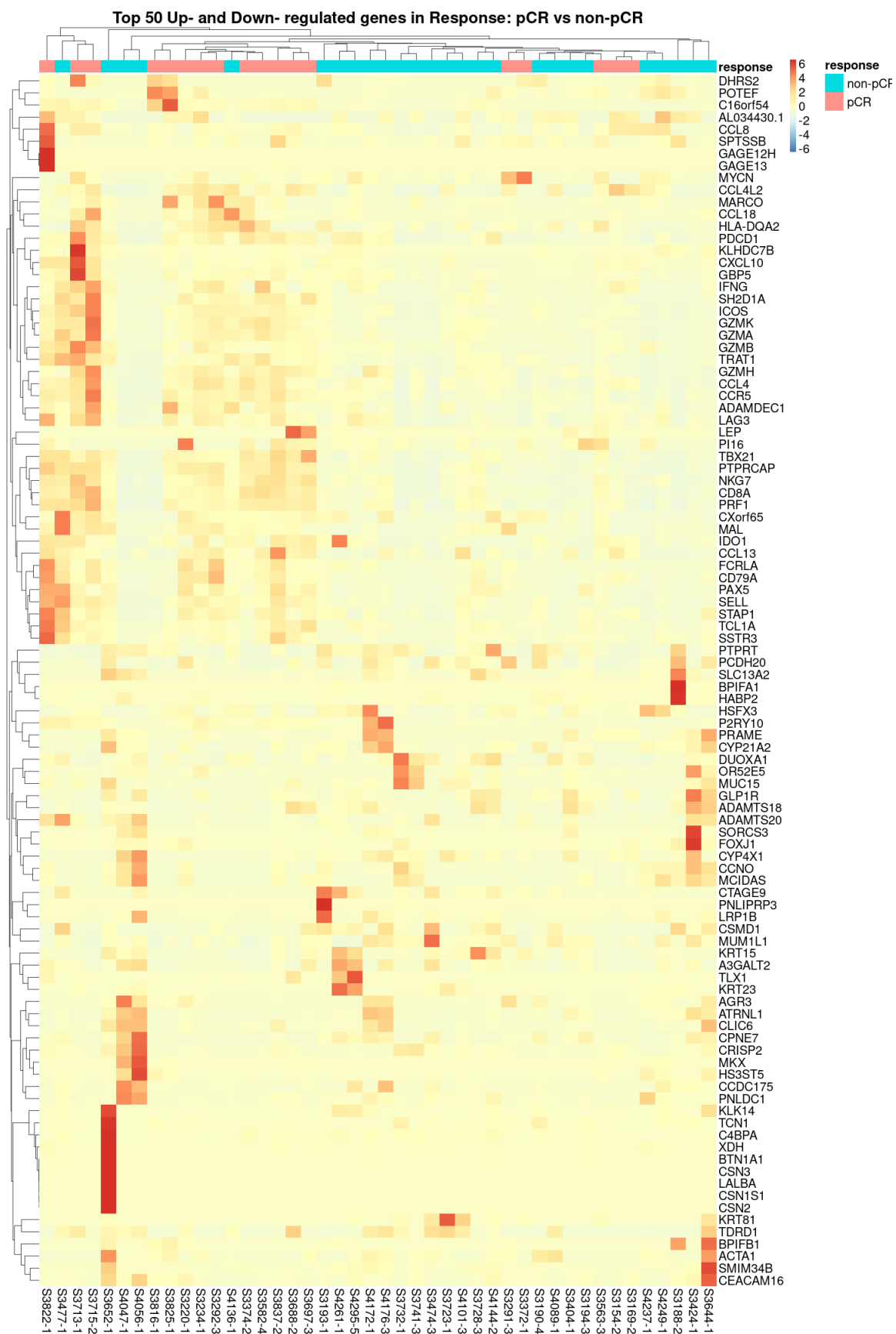
plotmat$ensembl_gene_id <- NULL

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)

```



```

# 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)

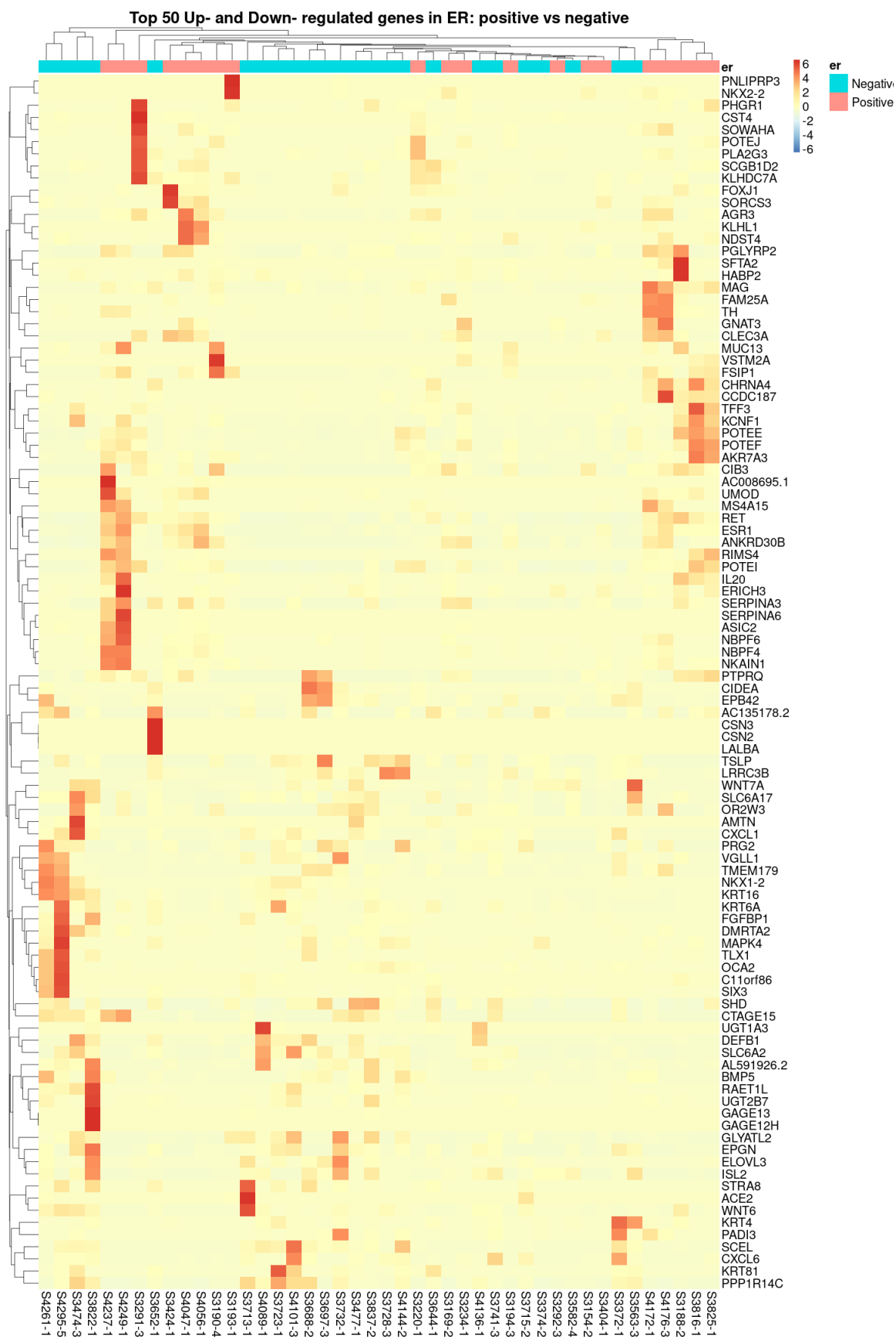
plotmat$ensembl_gene_id <- NULL

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)

```



```

# 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)

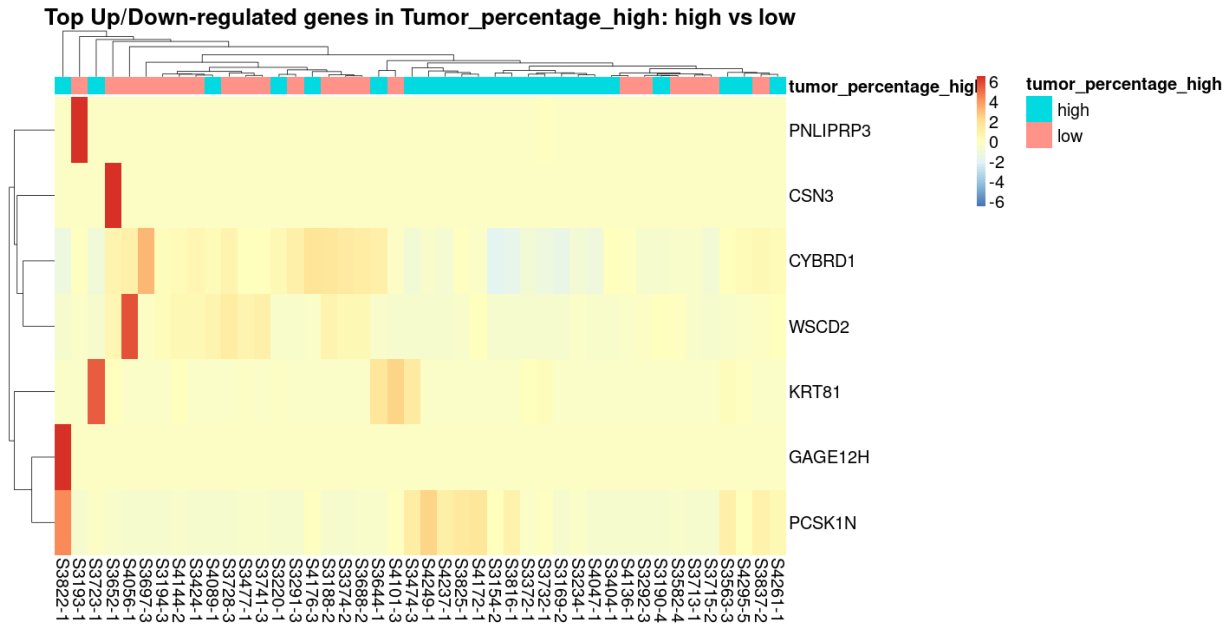
plotmat$ensembl_gene_id <- NULL

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)
```



```

# Create a matrix of normalized expression
sig_up <- resD0_tb_significant %>% arrange(-log2FoldChange) %>% head(50) %>% pull(gene)
sig_down <- resD0_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





## 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
## [27] haven_2.4.1
## [29] crayon_1.4.1
## [31] jsonlite_1.7.2
## [33] survival_3.2-7
## [35] gtable_0.3.0
## [37] XVector_0.30.0
## [39] GetoptLong_1.0.5
## [41] shape_1.4.5
## [43] scales_1.1.1
## [45] edgeR_3.32.1
## [47] progress_1.2.2
## [49] lasso2_1.2-21.1
## [51] clue_0.3-59
## [53] httr_1.4.2
## [55] farver_2.1.0
## [57] reshape_0.8.8
## [59] locfit_1.5-9.4
## [61] labeling_0.4.2
## [63] rlang_0.4.10
## [65] munsell_0.5.0
## [67] cellranger_1.1.0
## [69] cachem_1.0.4
## [71] generics_0.1.0
## [73] broom_0.7.6
## [75] fastmap_1.1.0
## [77] yaml_2.2.1
## [79] fs_1.5.0
## [81] quantreg_5.85
## [83] xml2_1.3.2
## [85] compiler_4.0.3
## [87] curl_4.3
## [89] interactiveDisplayBase_1.28.0
## [91] geneplotter_1.68.0
## [93] lattice_0.20-41
## [95] Matrix_1.2-18
## [97] vctrs_0.3.7
## [99] lifecycle_1.0.0
## [101] GlobalOptions_0.1.2
## [103] cowplot_1.1.1
## [105] rtracklayer_1.50.0
## [107] R6_2.5.0
## [109] MASS_7.3-53
## [111] openssl_1.4.3
## [113] withr_2.4.2
## [115] Rsamtools_2.6.0
## [117] GenomeInfoDbData_1.2.4
## [119] grid_4.0.3
## [121] Cairo_1.5-12.2
## [123] shiny_1.6.0

rappdirs_0.3.3
xfun_0.19
RCurl_1.98-1.3
genefilter_1.72.1
glue_1.4.2
zlibbioc_1.36.0
MatrixModels_0.5-0
DelayedArray_0.16.3
SparseM_1.81
DBI_1.1.1
Rcpp_1.0.6
xtable_1.8-4
tmvnsim_1.0-2
bit_4.0.4
ellipsis_0.3.1
pkgconfig_2.0.3
XML_3.99-0.6
utf8_1.2.1
tidyselect_1.1.0
later_1.2.0
BiocVersion_3.12.0
tools_4.0.3
cli_2.5.0
RSQLite_2.2.7
evaluate_0.14
ggdendro_0.1.22
bit64_4.0.5
nlme_3.1-149
mime_0.9
biomaRt_2.46.3
rstudioapi_0.13
png_0.1-7
reprex_2.0.0
stringi_1.5.3
ProtGenerics_1.22.0
psych_2.1.3
pillar_1.6.0
BiocManager_1.30.12
conquer_1.0.2
bitops_1.0-7
httpuv_1.6.0
promises_1.2.0.1
assertthat_0.2.1
rjson_0.2.20
GenomicAlignments_1.26.0
mnormt_2.0.2
hms_1.0.0
rmarkdown_2.5
logging_0.10-108
lubridate_1.7.10

```