DE analysis

Sergey Naumenko

2021-05-10

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

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

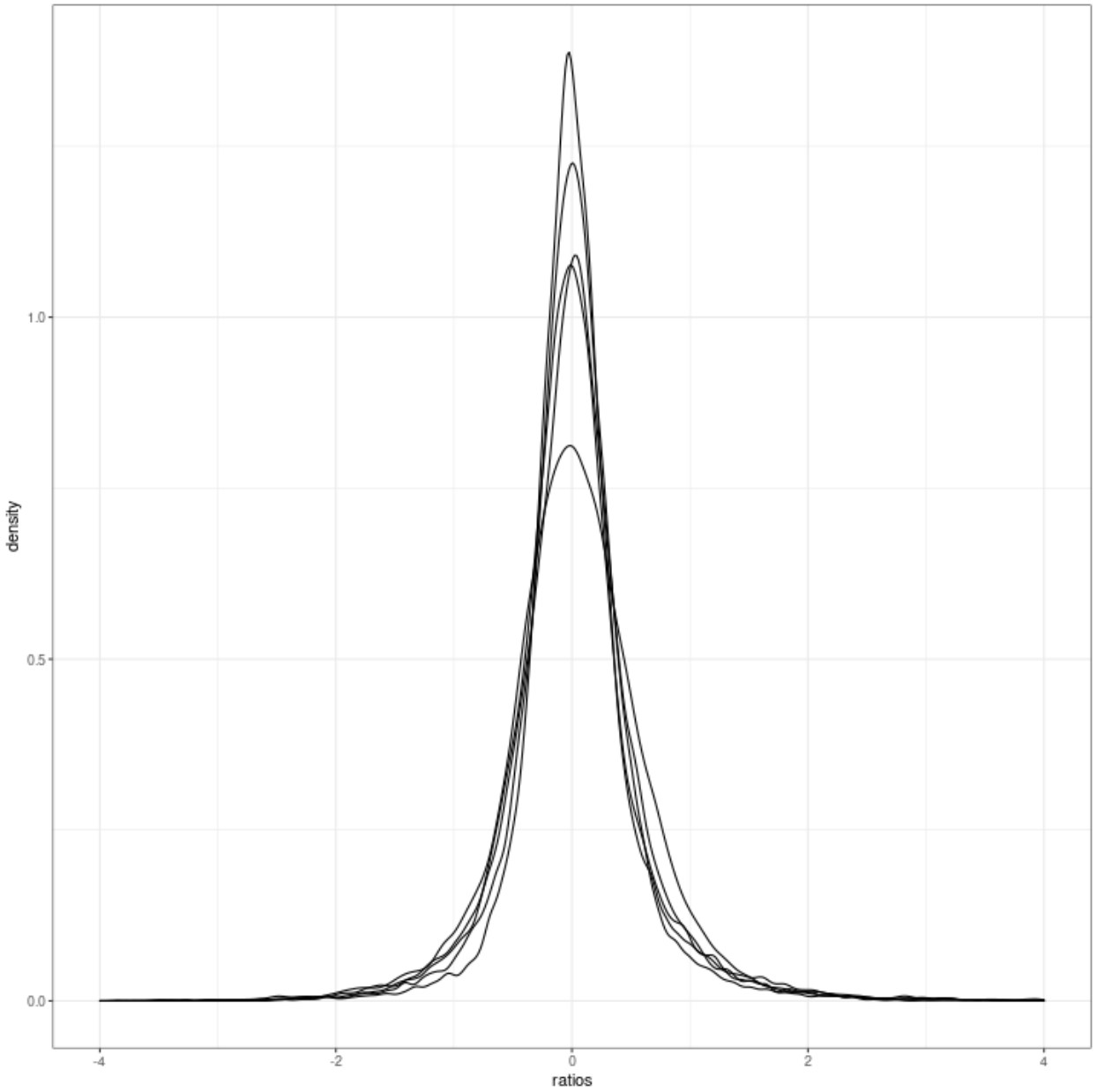
Wald test

Here we subset protein coding genes.

DEGreport QC

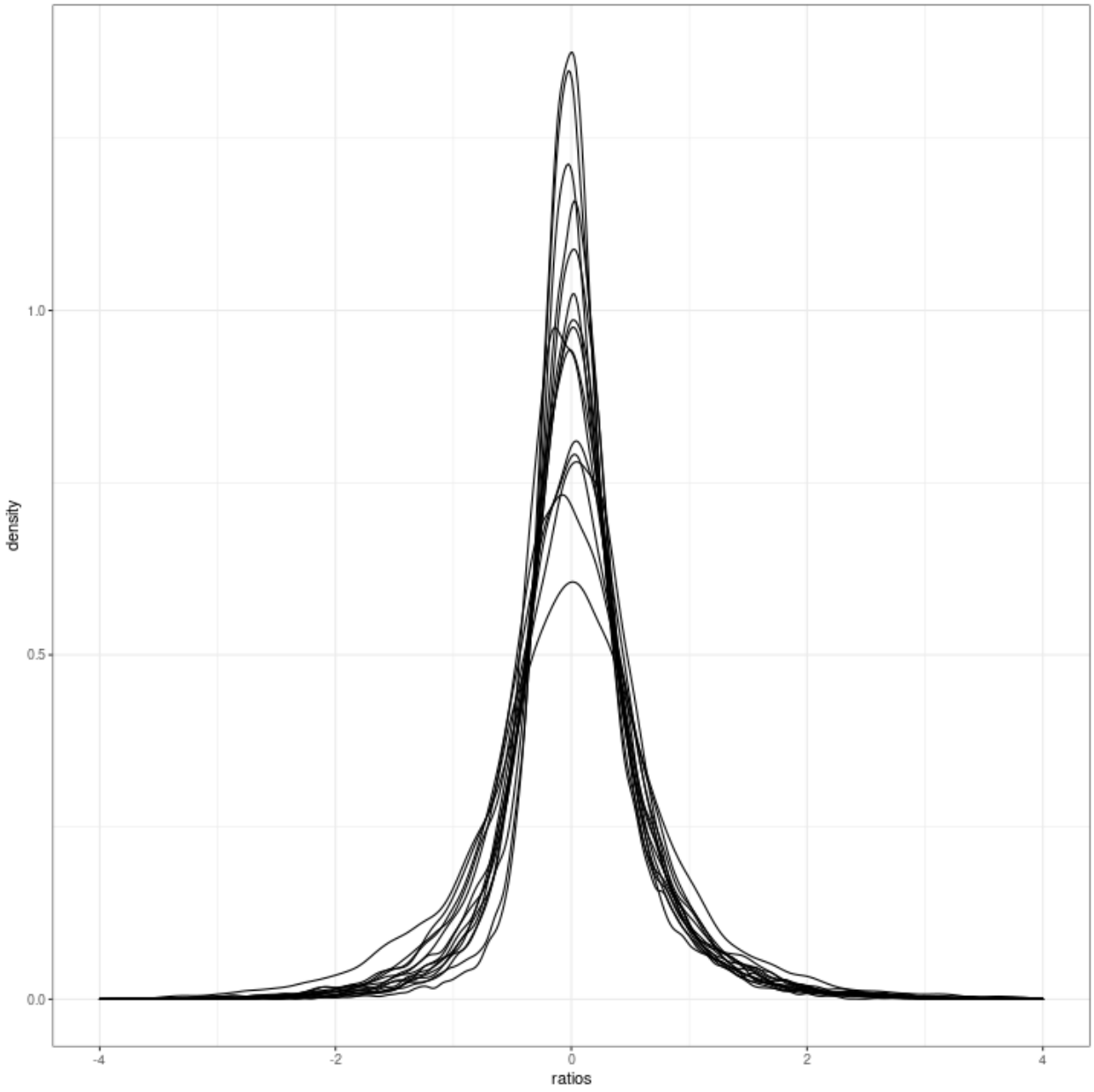
Size factor QC - samples 1-15

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



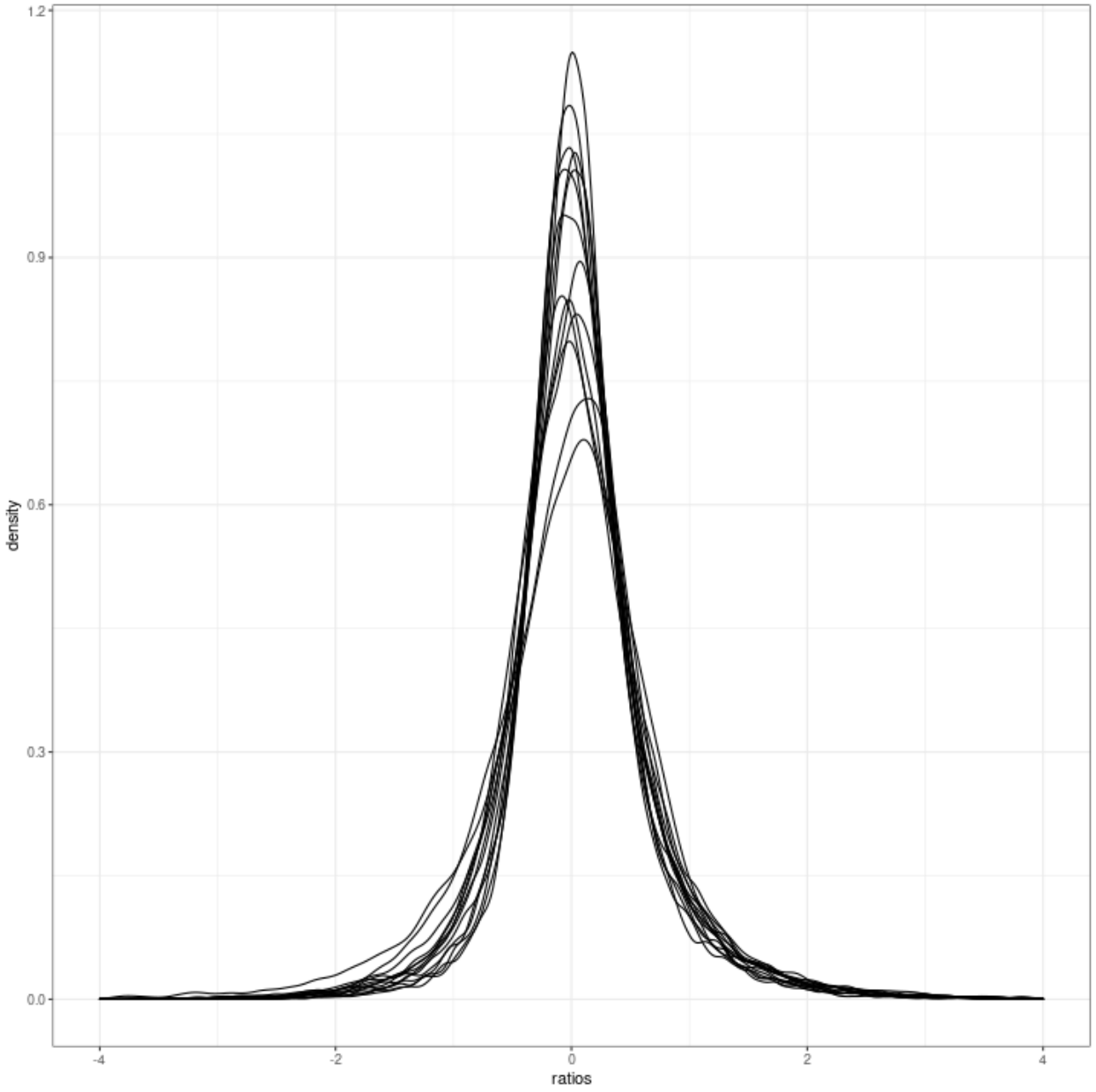
Size factor QC - samples 16-30

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



Size factor QC - samples 31-40 (44)

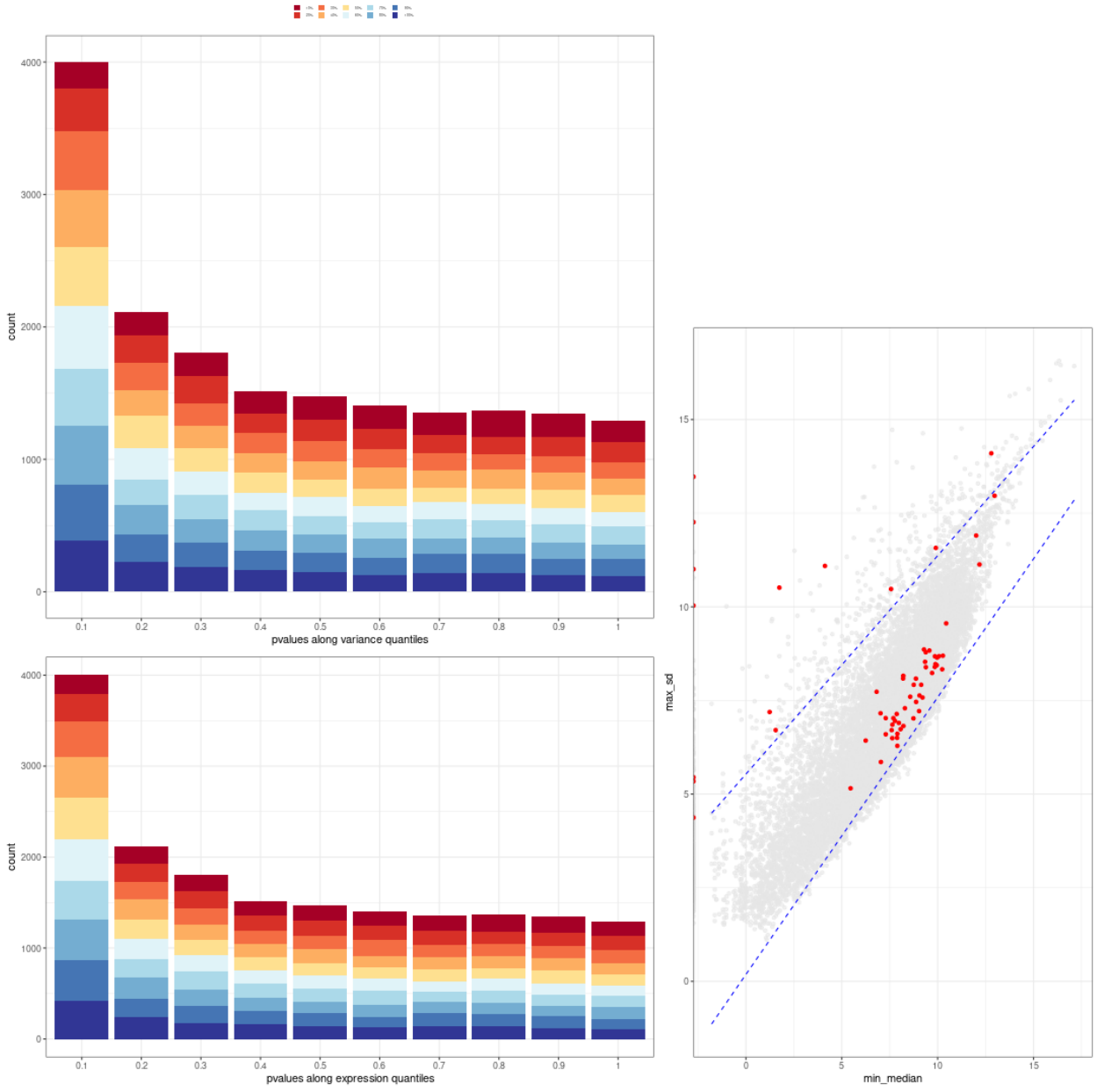
```
degCheckFactors(counts[, 31:ncol(counts)])
```



Mean-Variance QC plots

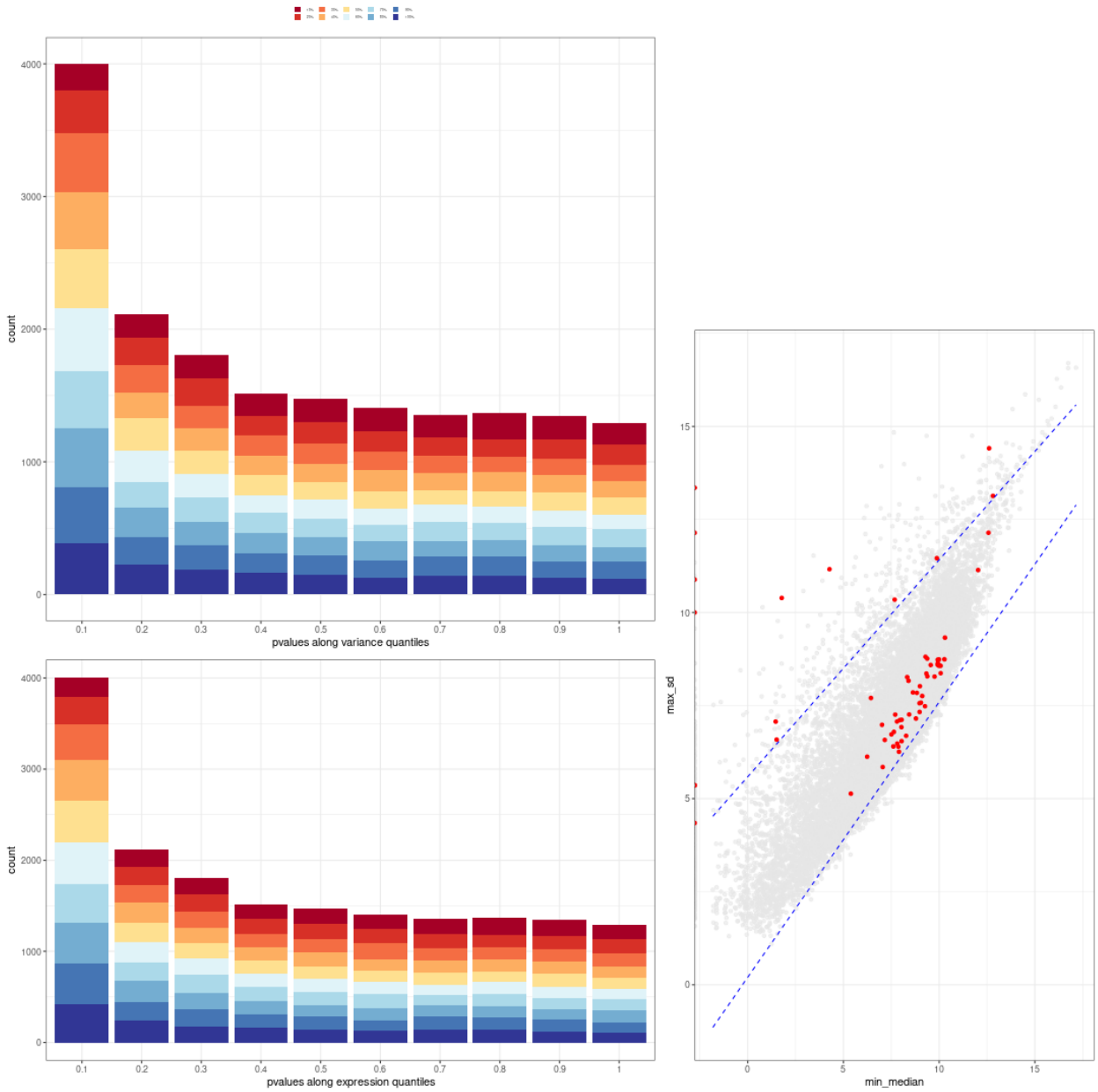
treatment

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



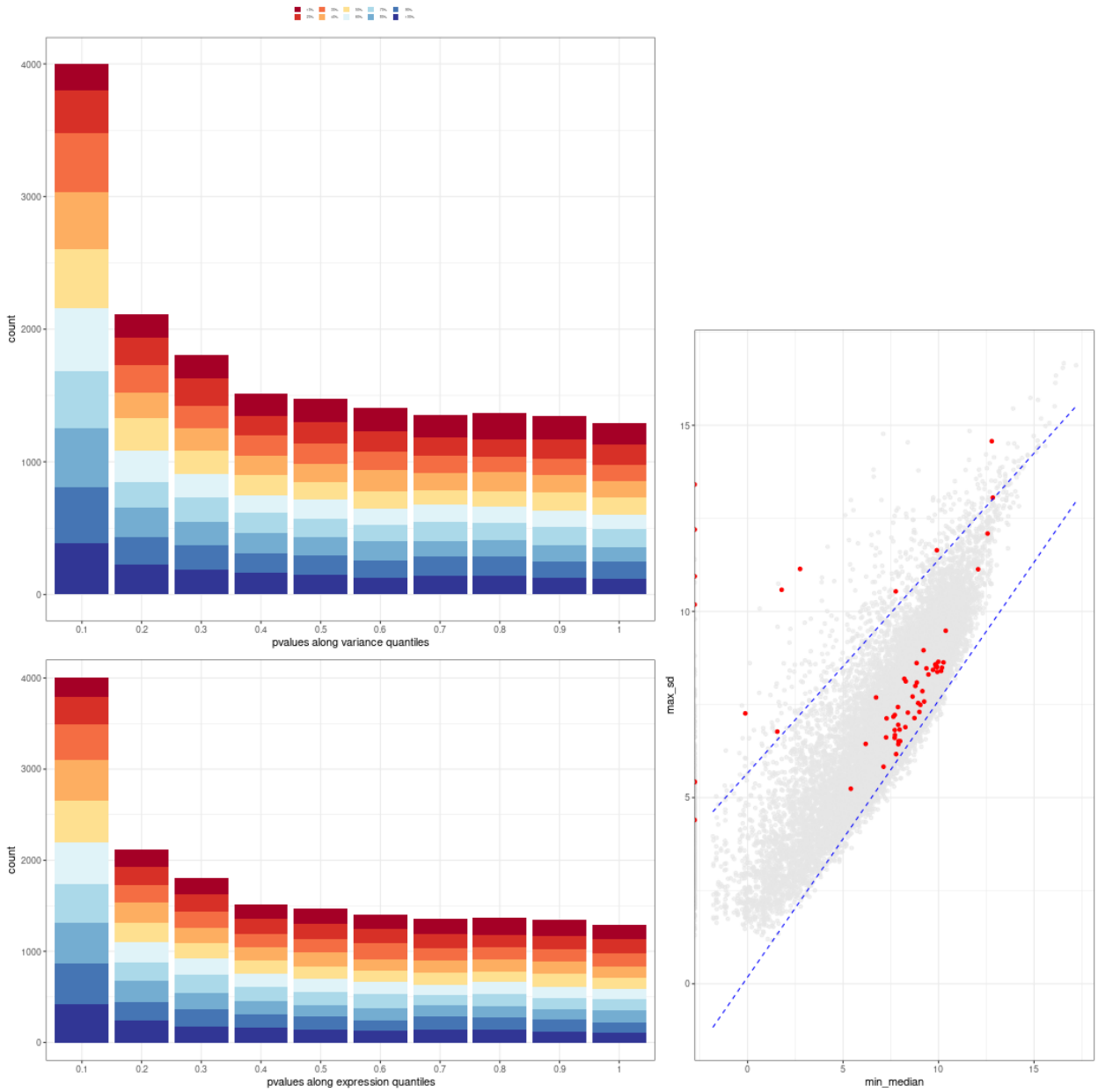
response

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



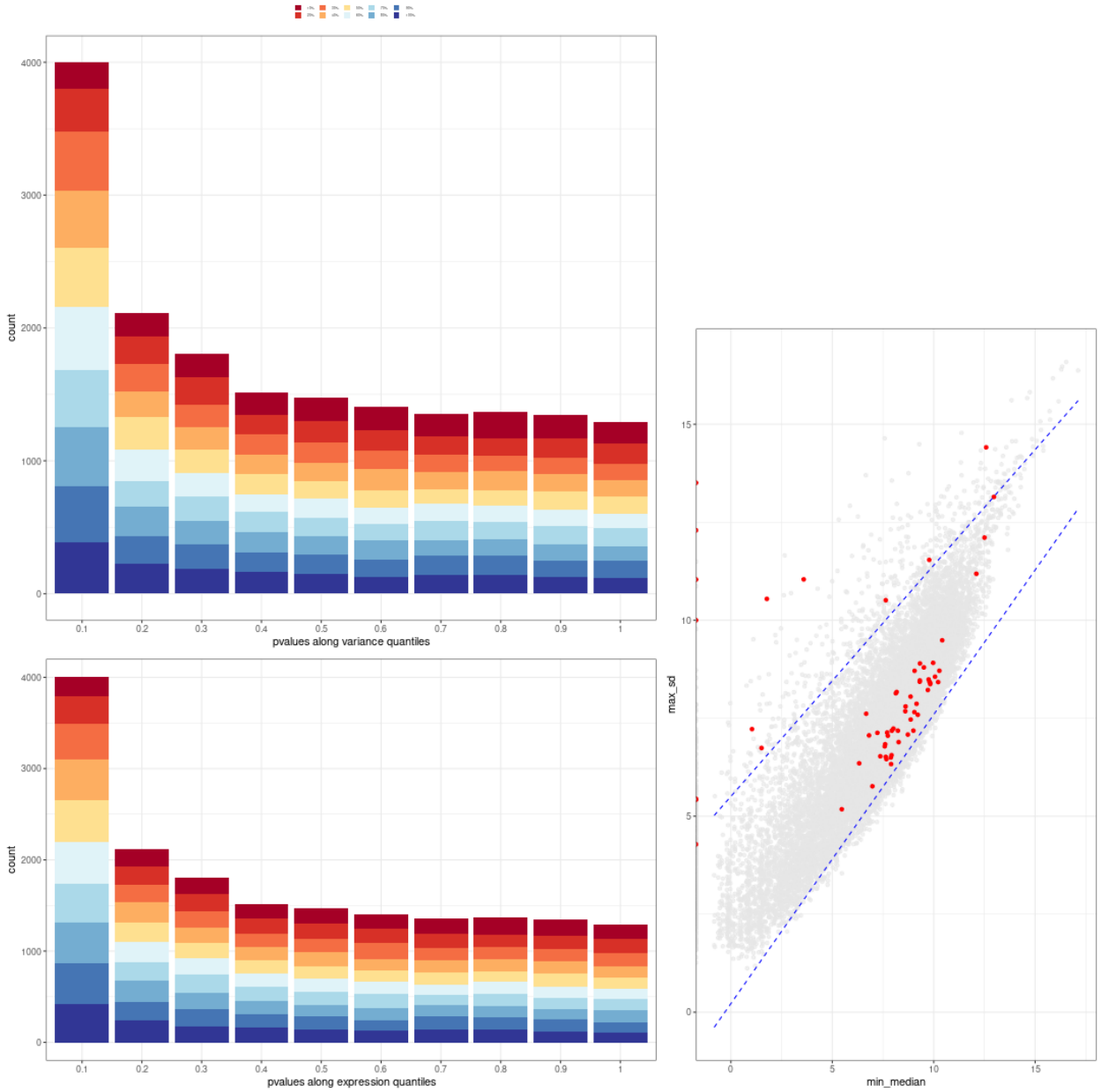
ER

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



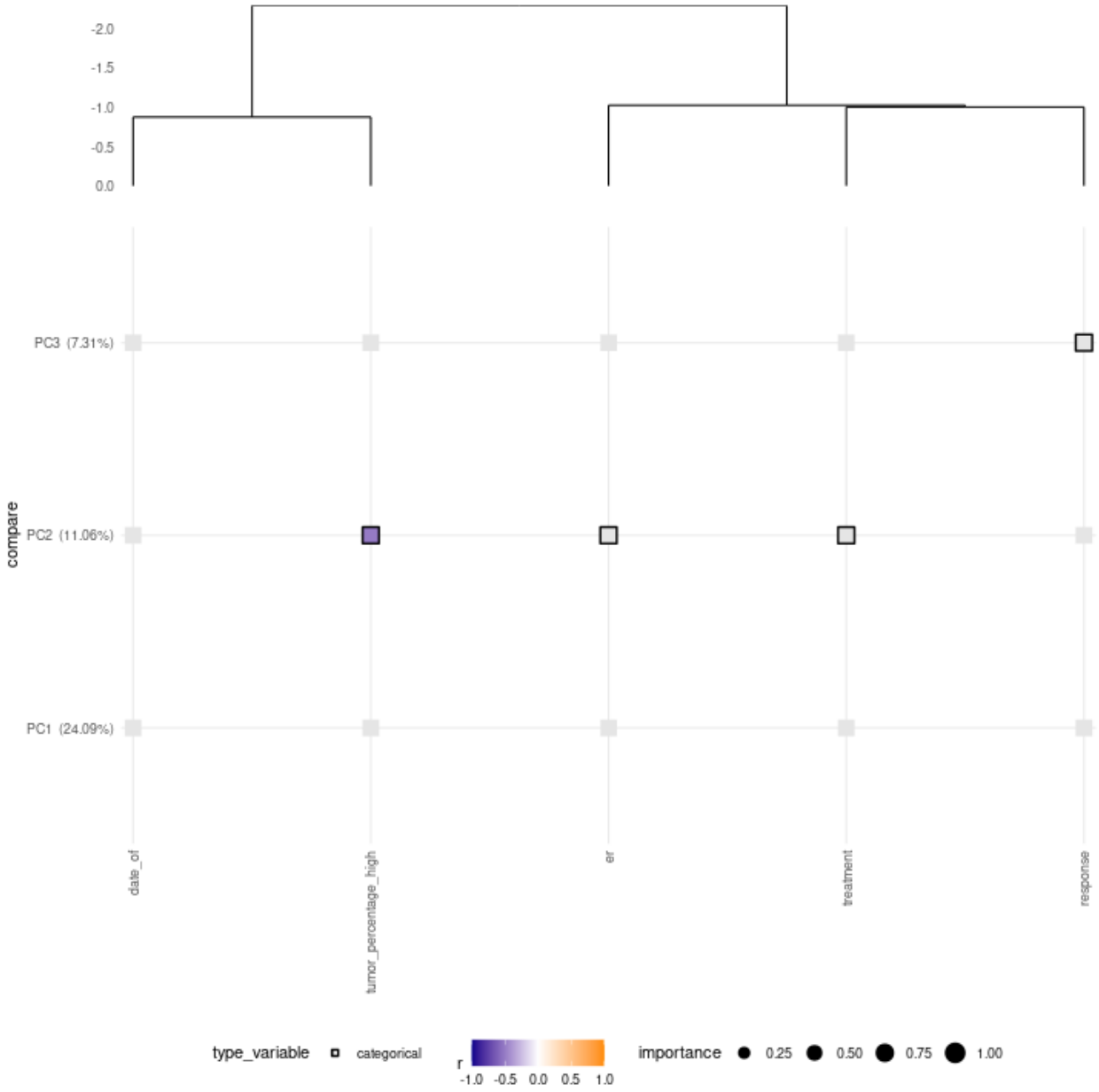
tumor_percentage_high

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

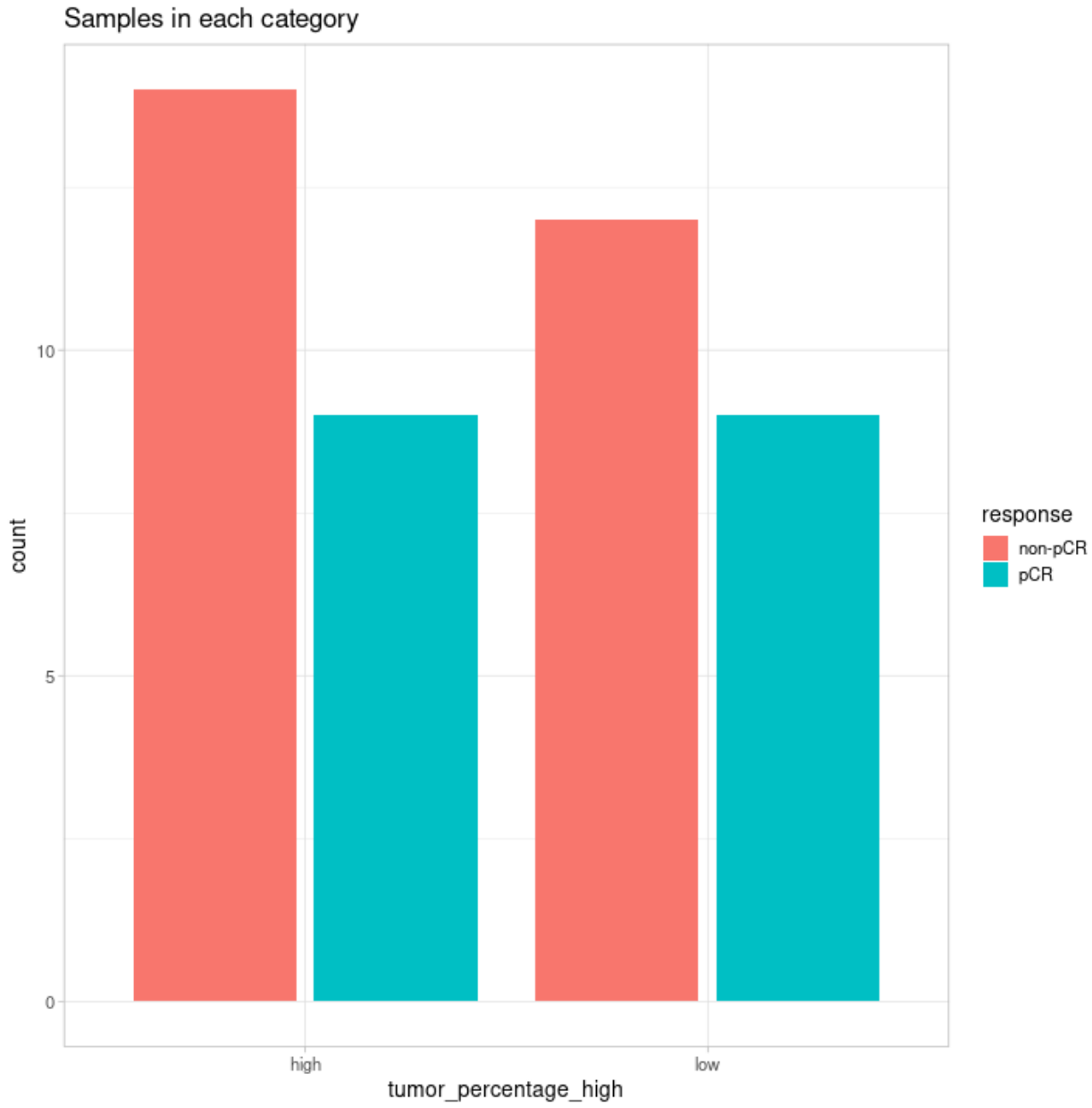


Covariates effect on count data

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



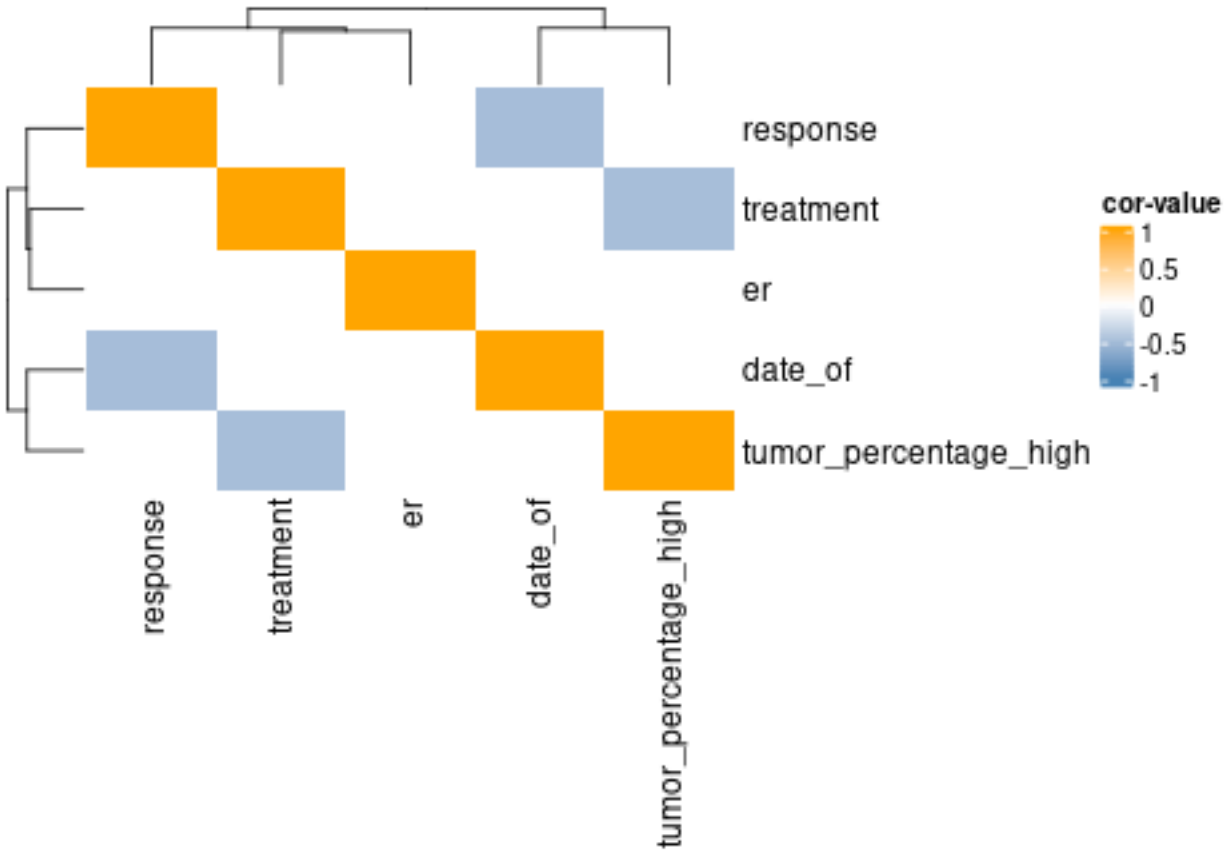
```
mdata %>% ggplot(aes(tumor_percentage_high, fill = response)) +
  geom_bar(position = "dodge2") +
  ggtitle("Samples in each category")
```



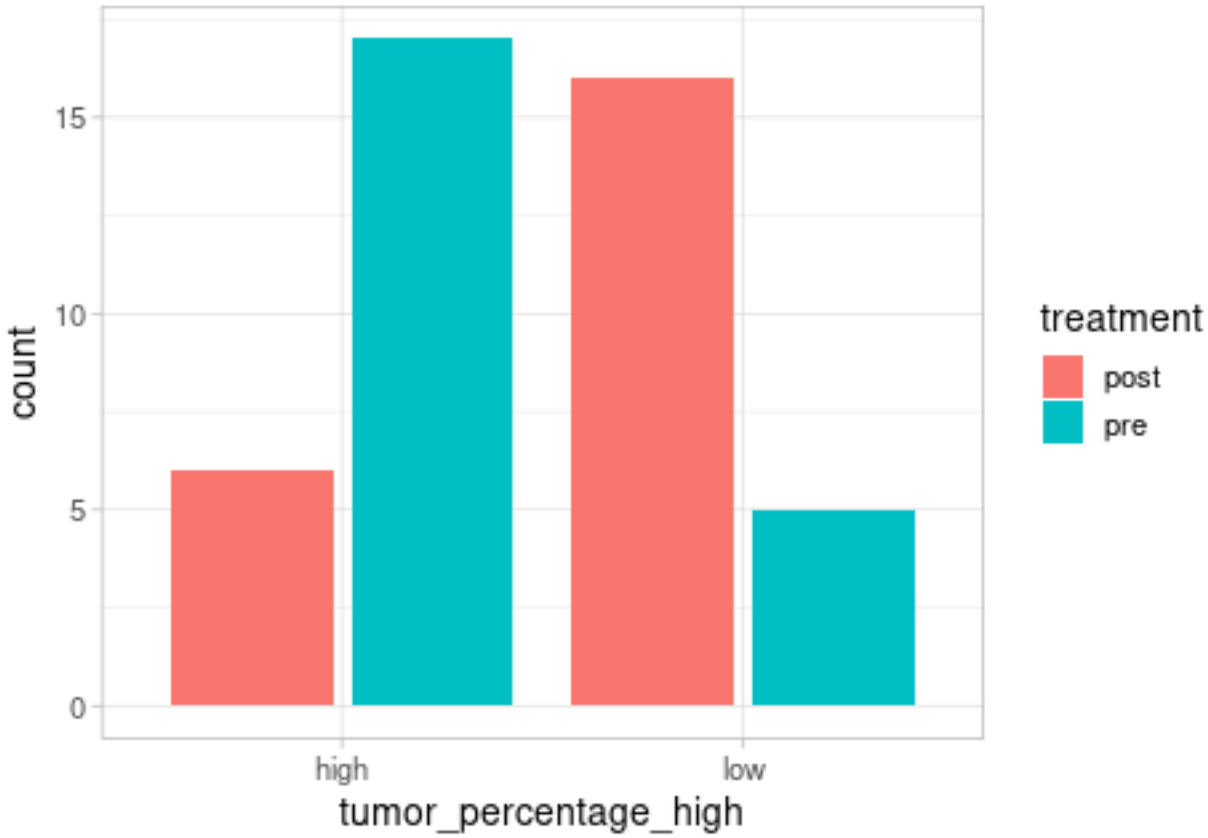
Samples split equally between tumor_percentage_high for both response types. That allows to control for tumor_percentage_high batch effectively.

Covariates correlation with metrics

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

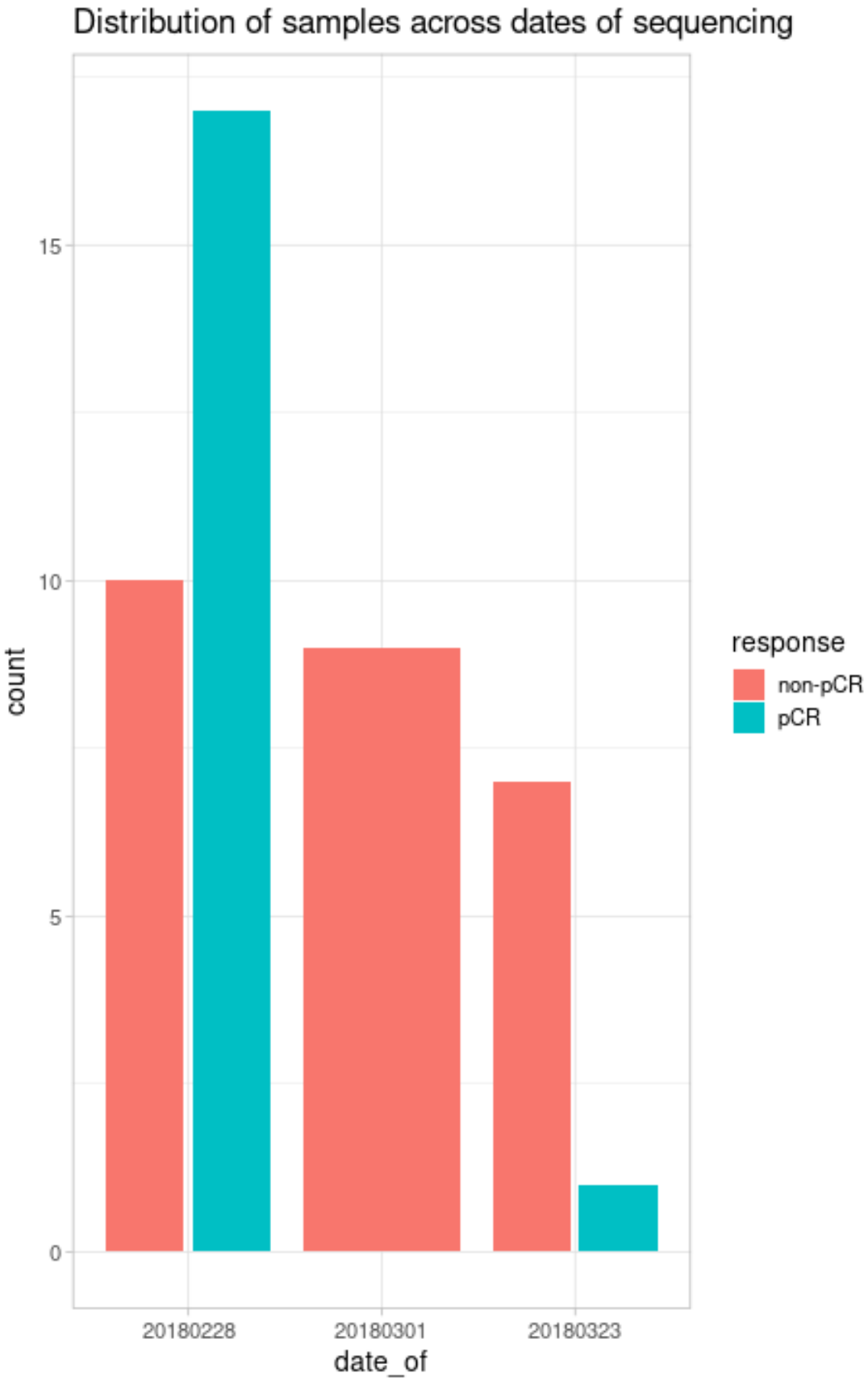


```
mdata %>% ggplot(aes(tumor_percentage_high, fill = treatment)) + geom_bar(position = "dodge2")
```



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

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

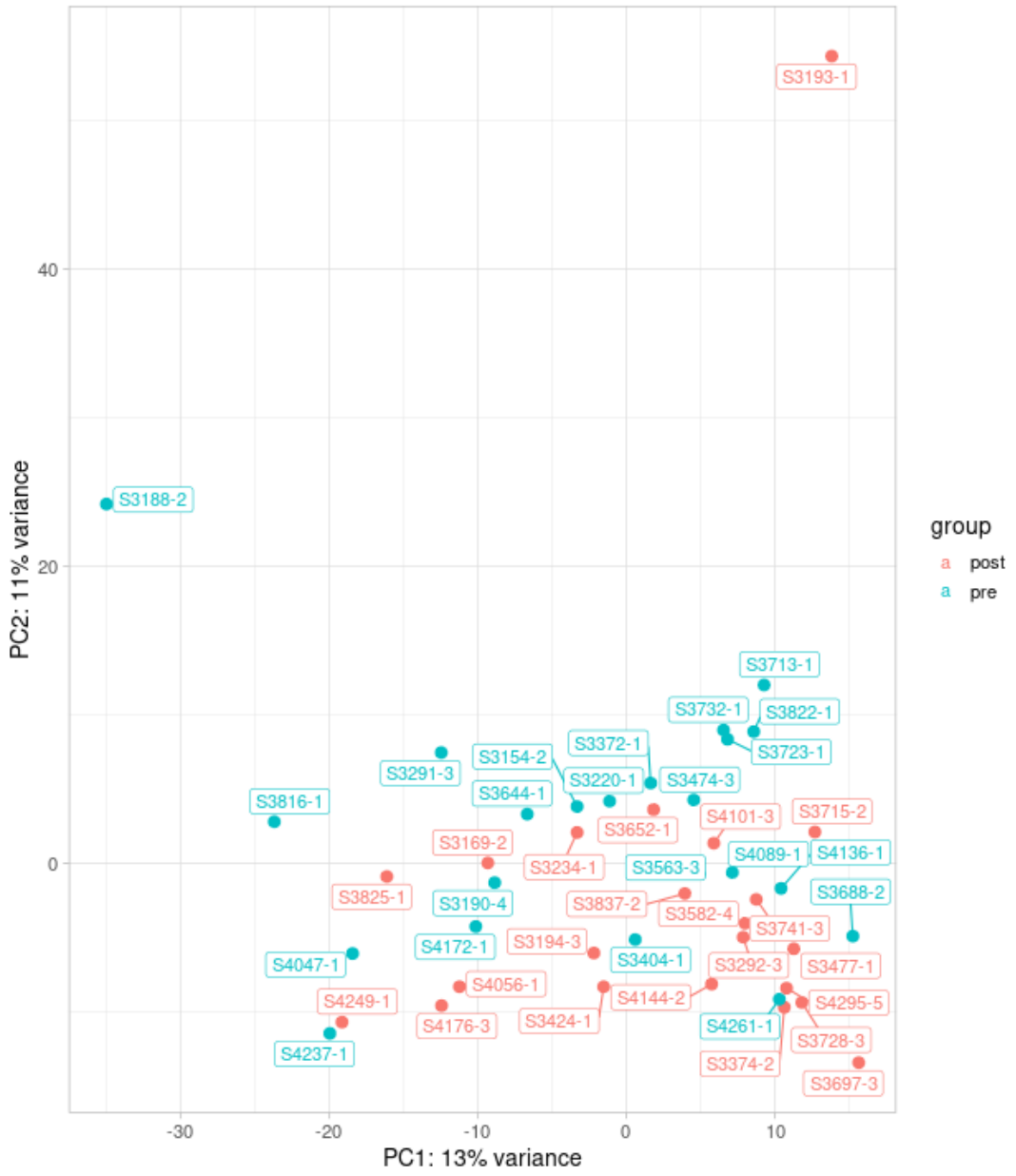

Distribution of samples across dates of sequencing
count
15
10
5
0
20180228
20180301
20180323
date_of
response
non-pCR
pCR

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

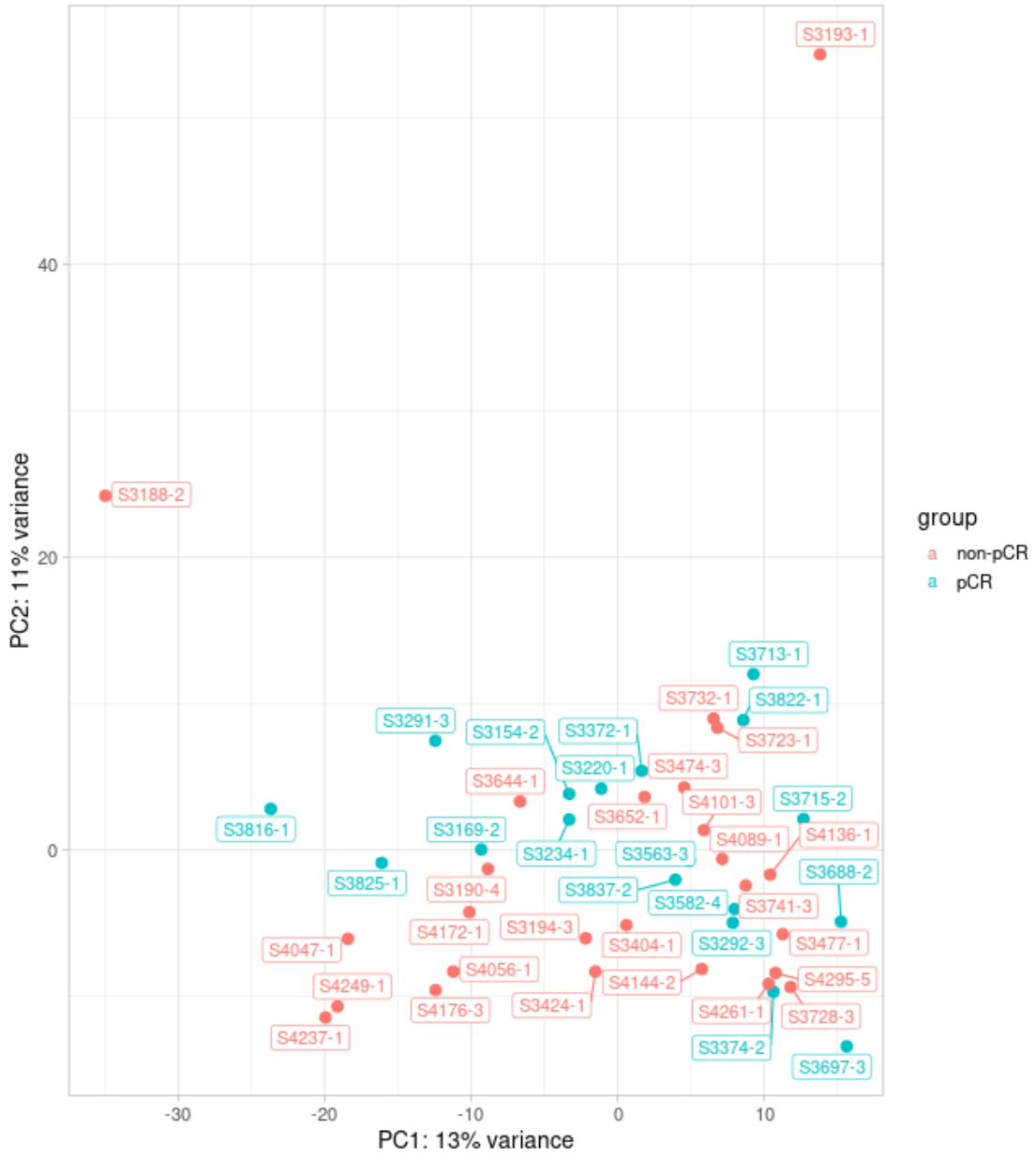
PCA - treatment

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



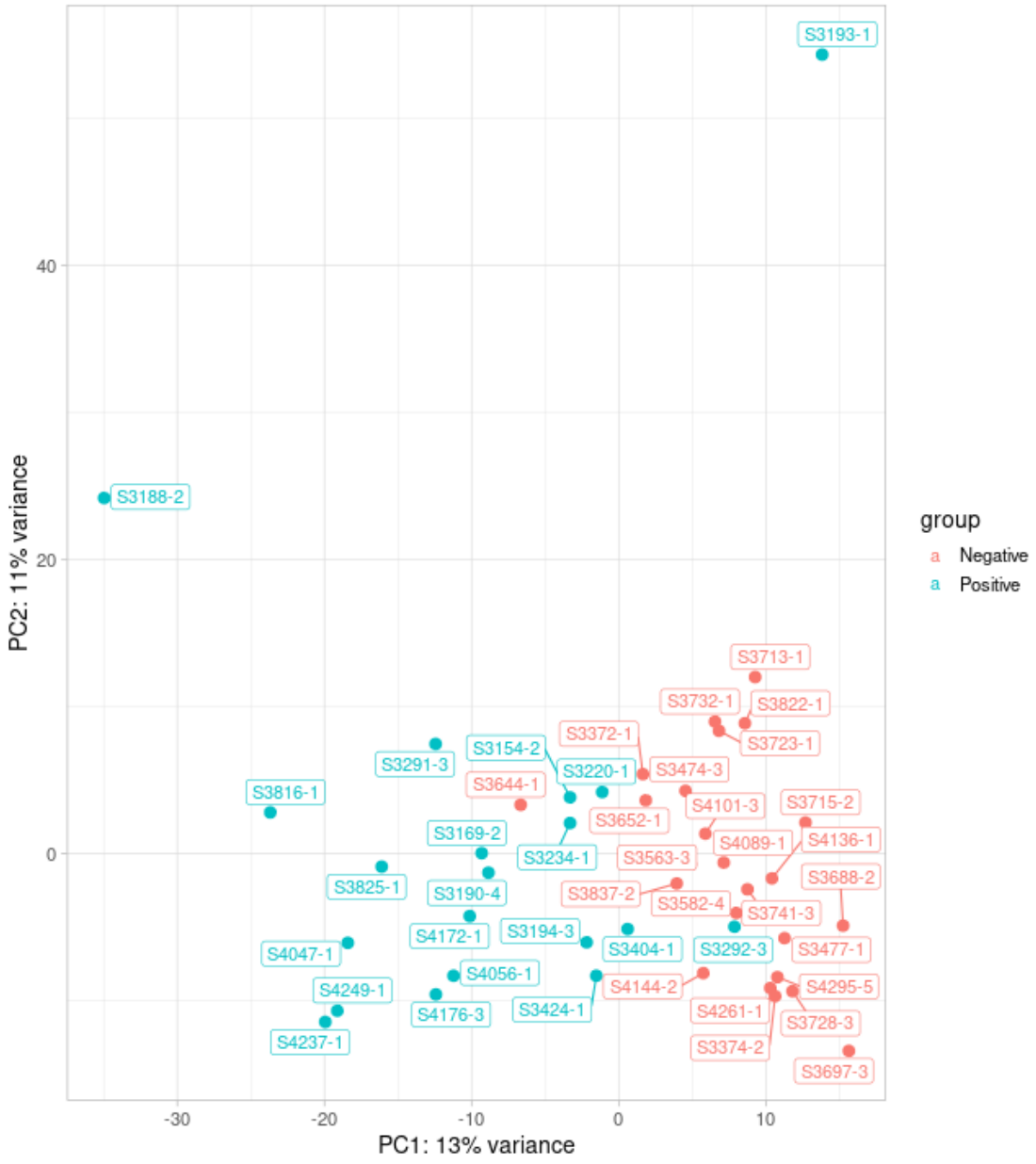
PCA - response

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



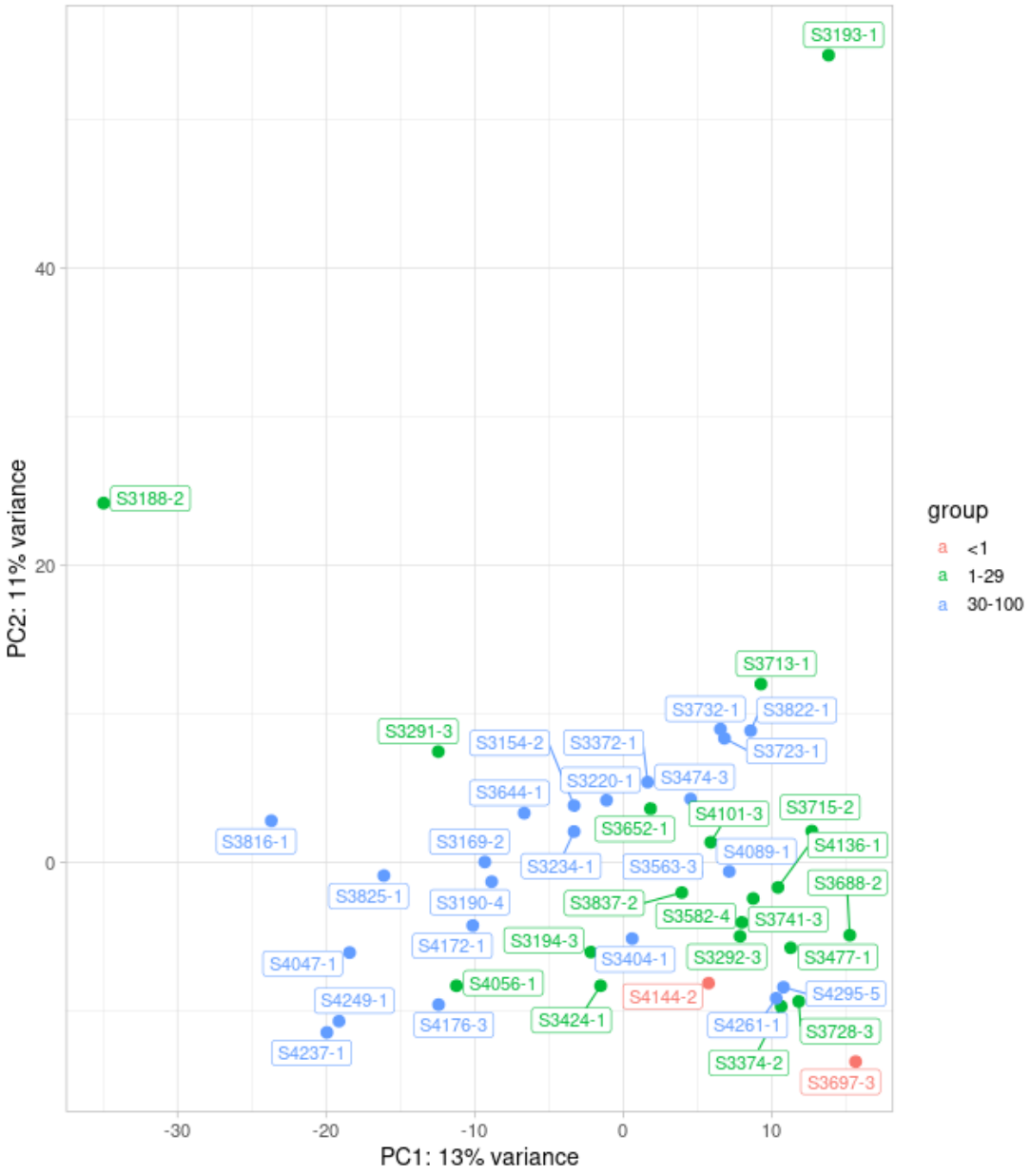
PCA - ER

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



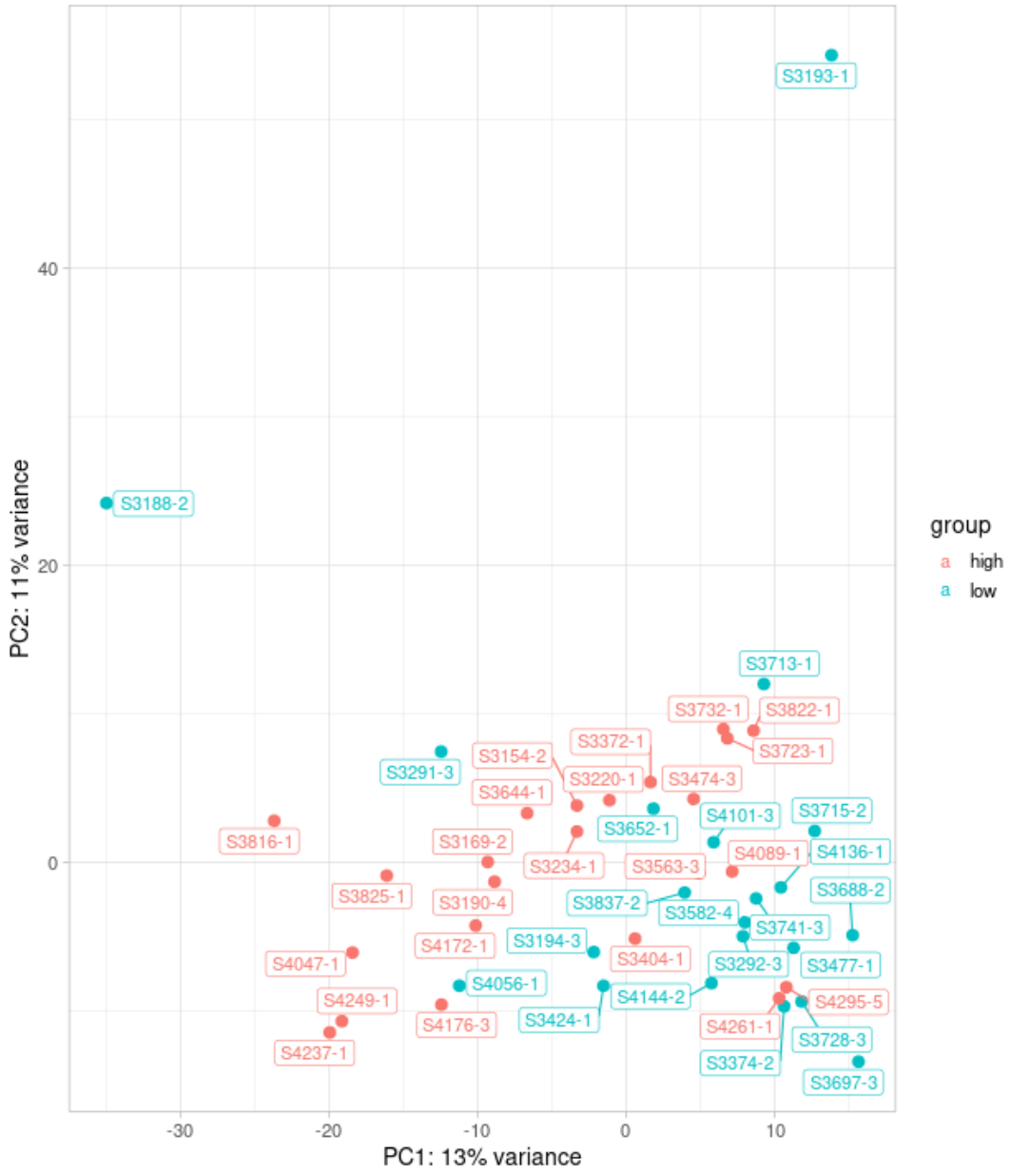
PCA - tumor_percentage

```
# 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

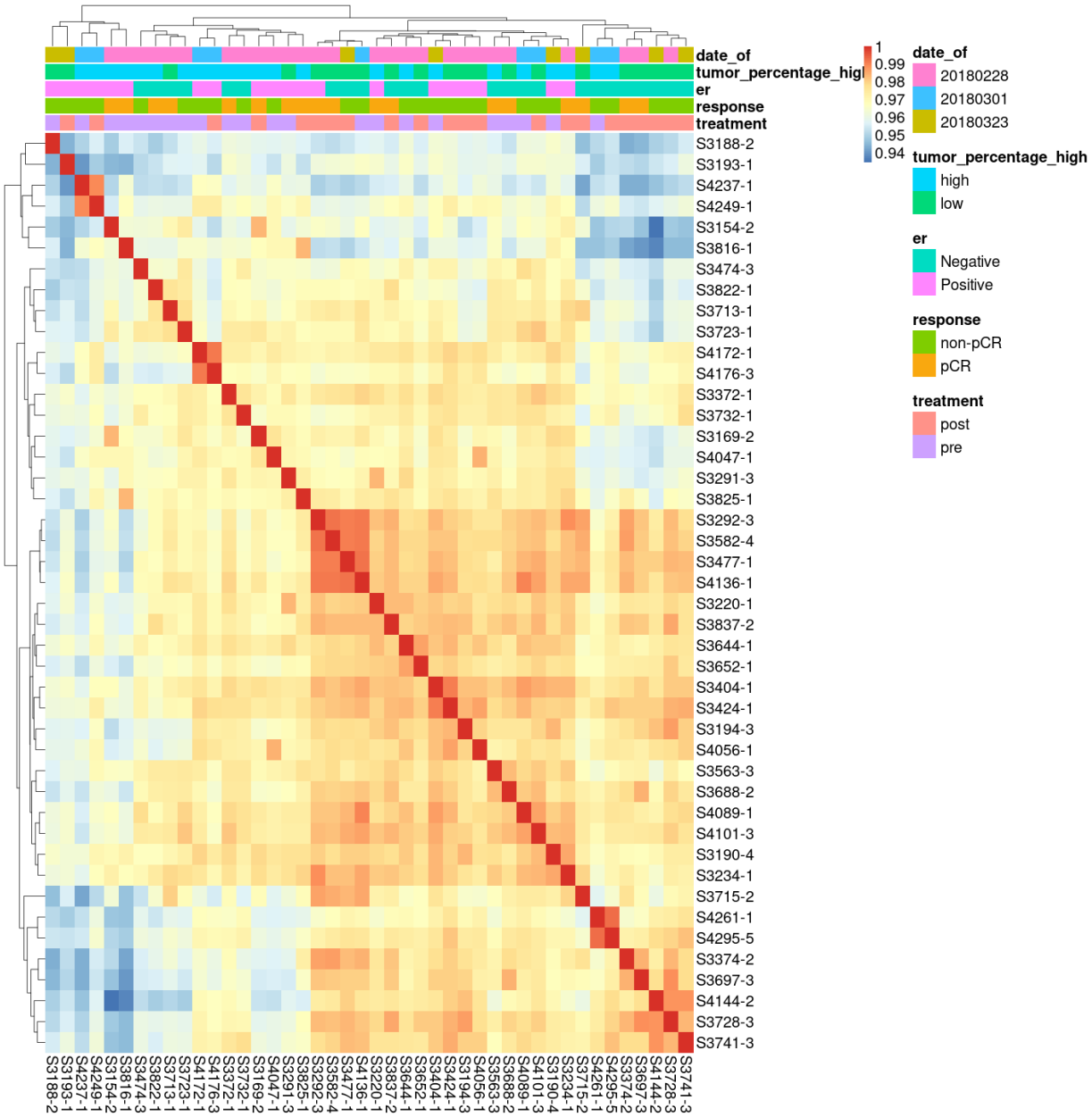
```
# Use the DESeq2 function
plotPCA(rld, intgroup = c("date_of"))  + geom_label_repel(aes(label = name))
```


S3193-1
S3188-2
S3713-1
S3732-1
S3822-1
S3291-3
S3154-2
S3372-1
S3723-1
S3220-1
S3474-3
S3644-1
S3816-1
S4101-3
S3715-2
S3652-1
S3169-2
S4089-1
S4136-1
S3234-1
S3563-3
S3825-1
S3688-2
S3190-4
S3837-2
S3582-4
S3741-3
S4047-1
S4172-1
S3194-3
S3404-1
S3292-3
S3477-1
S4056-1
S4144-2
S4295-5
S4249-1
S4176-3
S3424-1
S4261-1
S3728-3
S4237-1
S3374-2
S3697-3
40
20
0
-30
-20
-10
0
10
PC2: 11% variance
PC1: 13% variance
group
a 20180228
a 20180301
a 20180323

Inter-correlation analysis

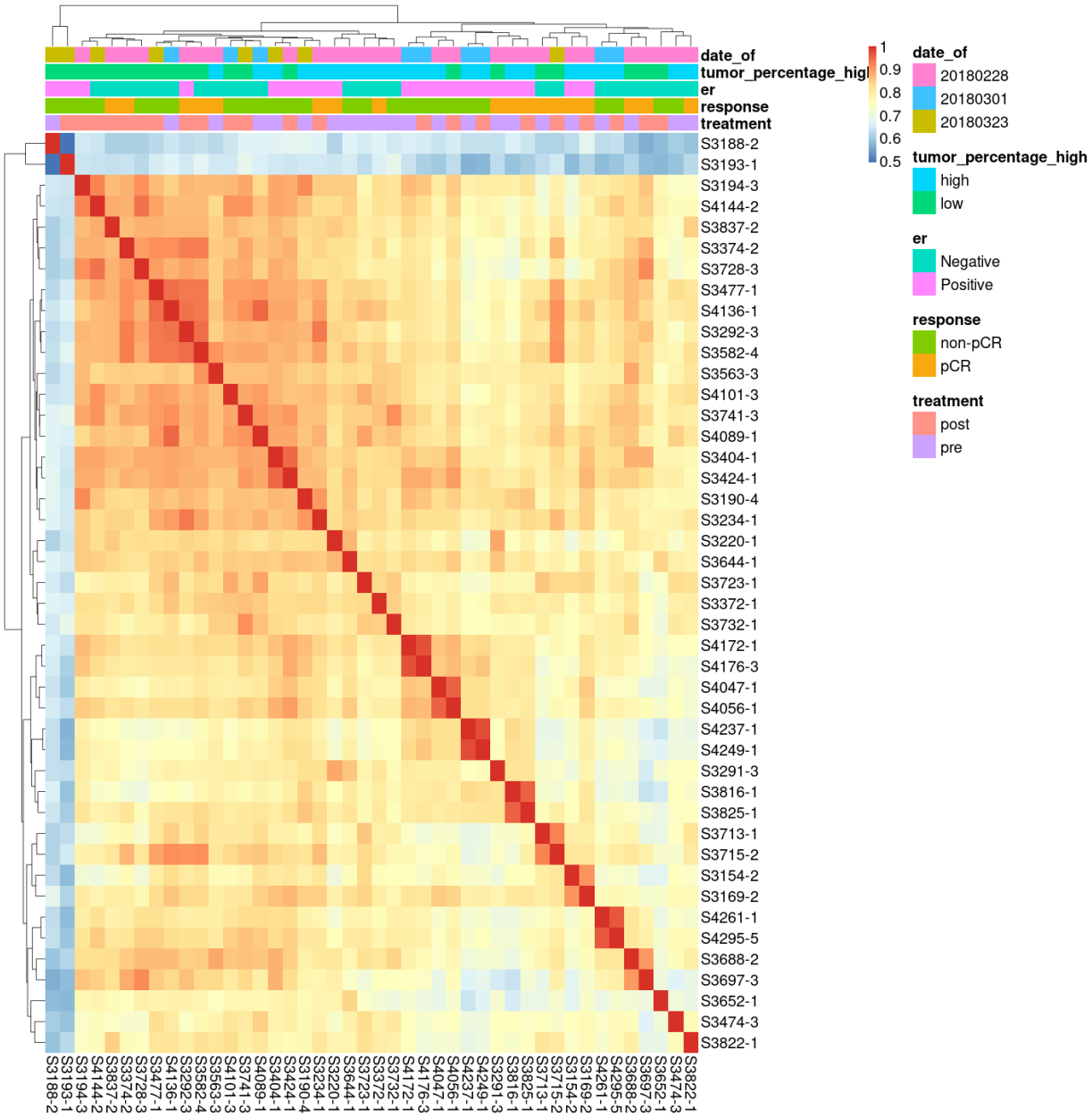
Without study_id

```
# 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)
```


date_of
tumor_percentage_high
er
response
treatment
S3188-2
S3193-1
S4237-1
S4249-1
S3154-2
S3816-1
S3474-3
S3822-1
S3713-1
S3723-1
S4172-1
S3176-3
S3372-1
S3732-1
S3169-2
S4047-1
S3291-3
S3825-1
S3292-3
S3582-4
S3477-1
S4136-1
S3220-1
S3837-2
S3644-1
S3652-1
S3404-1
S3424-1
S3194-3
S4056-1
S3563-3
S3688-2
S4089-1
S4101-3
S3190-4
S3234-1
S3715-2
S4261-1
S4295-5
S3374-2
S3697-3
S4144-2
S3728-3
S3741-3
1
0.99
0.98
0.97
0.96
0.95
0.94
date_of
20180228
20180301
20180323
tumor_percentage_high
high
low
er
Negative
Positive
response
non-pCR
pCR
treatment
post
pre

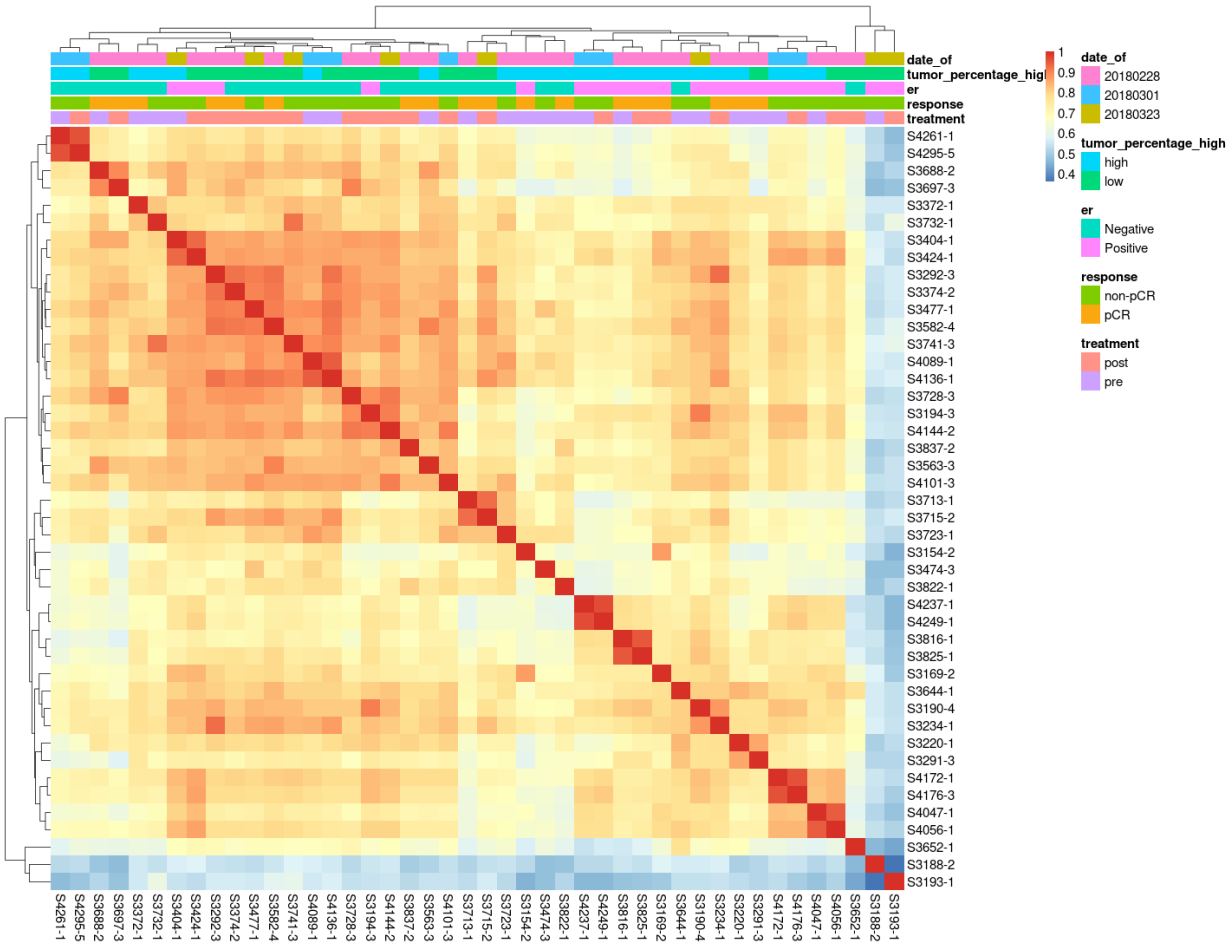
Without study_id = top 1000 variable genes

```
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)
```

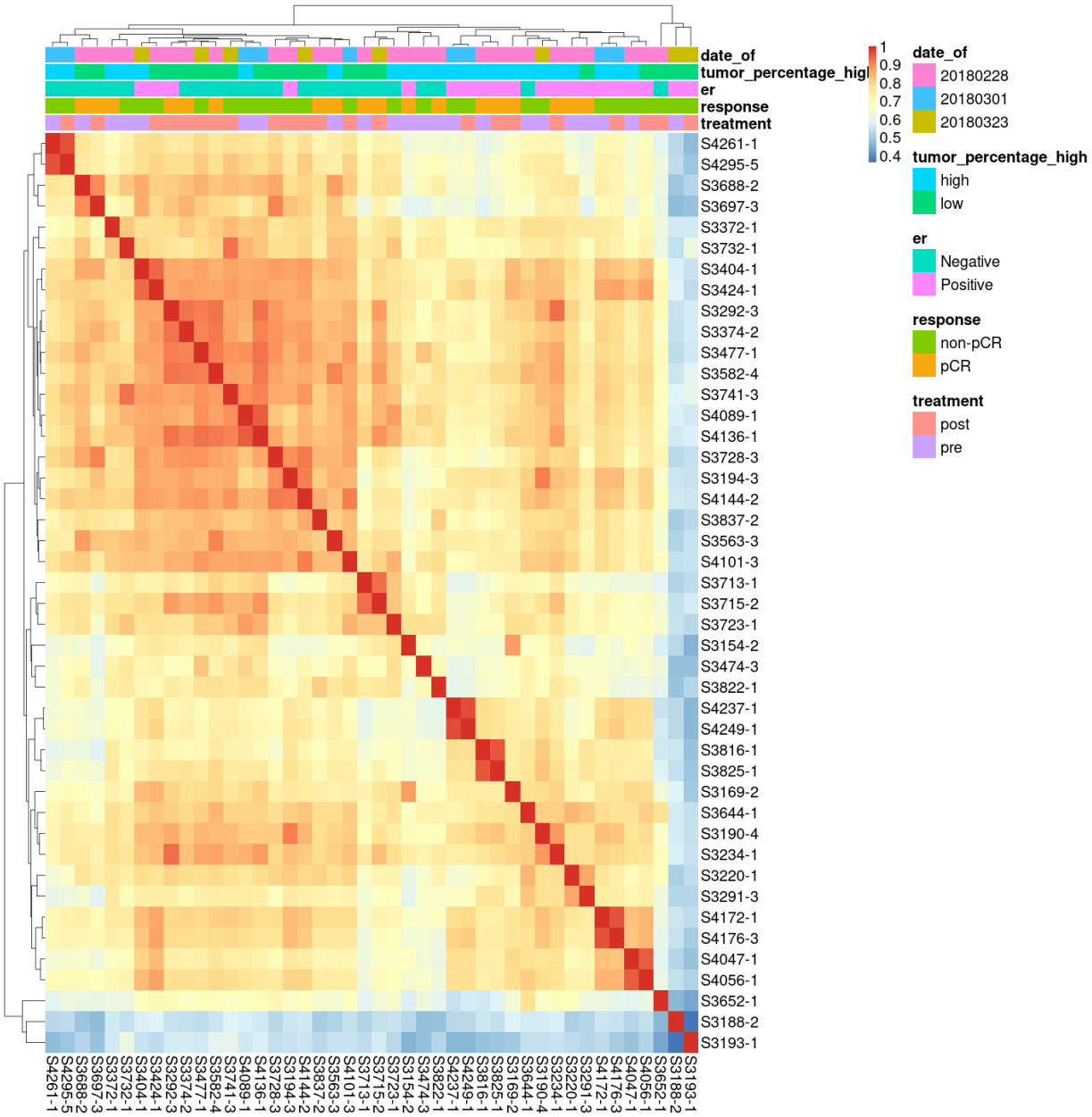

date_of
tumor_percentage_high
er
response
treatment
S3188-2
S3193-1
S3194-3
S4144-2
S3837-2
S3374-2
S3728-3
S3477-1
S4136-1
S3292-3
S3582-4
S3563-3
S4101-3
S3741-3
S4089-1
S3404-1
S3424-1
S3190-4
S3234-1
S3220-1
S3644-1
S3723-1
S3372-1
S3732-1
S4172-1
S4176-3
S4047-1
S4056-1
S4237-1
S4249-1
S3291-3
S3816-1
S3825-1
S3713-1
S3715-2
S3154-2
S3169-2
S4261-1
S4295-5
S3688-2
S3697-3
S3652-1
S3474-3
S3822-1
1
0.9
0.8
0.7
0.6
0.5
date_of
20180228
20180301
20180323
tumor_percentage_high
high
low
er
Negative
Positive
response
non-pCR
pCR
treatment
post
pre

Wihout 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
fig3a <- as.ggplot(pheatmap(rld_cor,
         annotation = annotation,
         border = NA,
         fontsize = 15,
         cellheight = 20))
```

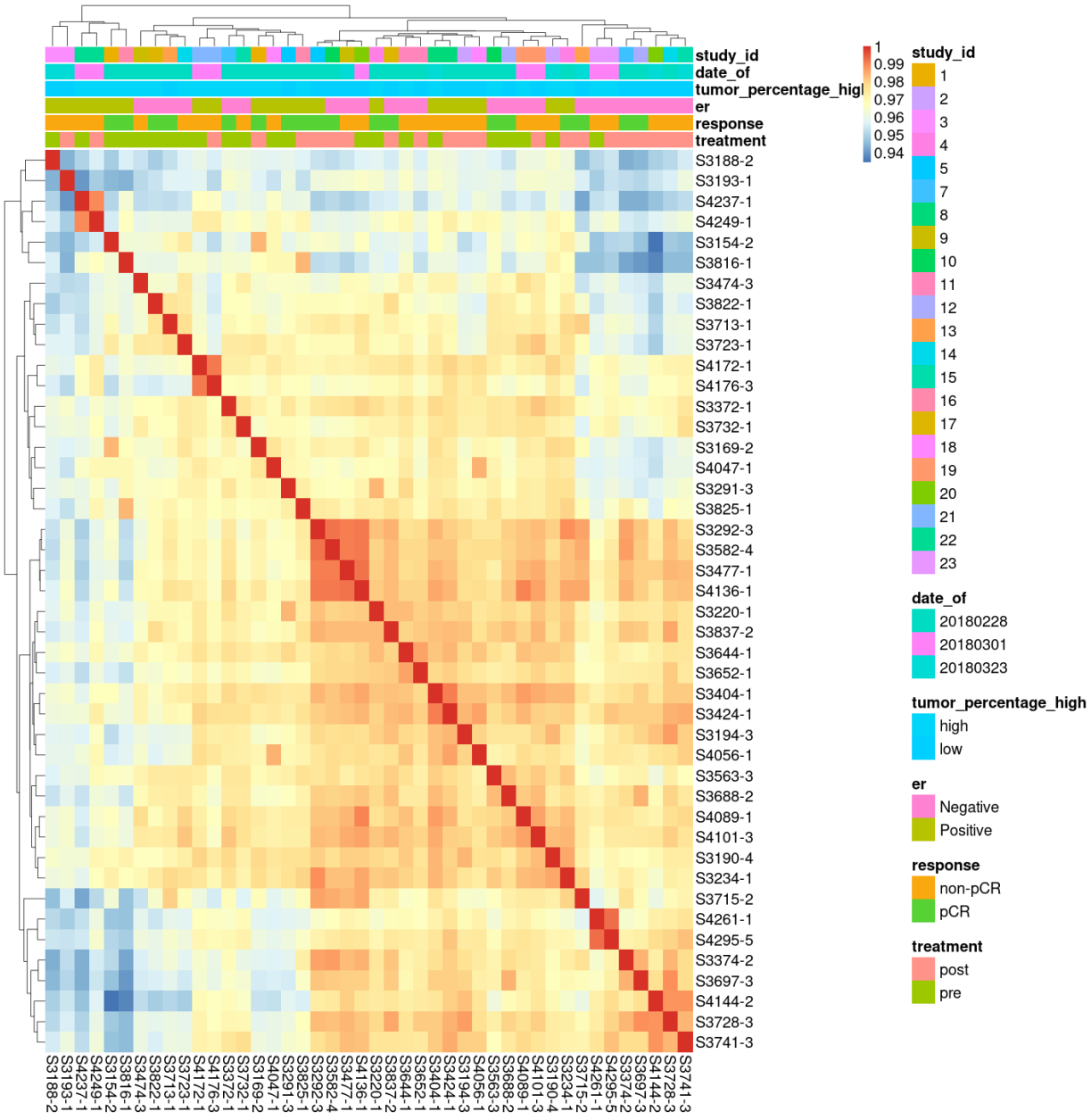



```
saveRDS(fig3a, "data/fig3a.RDS")
pheatmap(rld_cor,
         annotation = annotation,
         border = NA,
         fontsize = 20)
```


date_of
tumor_percentage_high
er
response
treatment
S4261-1
S4295-5
S3688-2
S3697-3
S3372-1
S3732-1
S3404-1
S3424-1
S3292-3
S3374-2
S3477-1
S3582-4
S3741-3
S4089-1
S4136-1
S3728-3
S3194-3
S4144-2
S3837-2
S3563-3
S4101-3
S3713-1
S3715-2
S3723-1
S3154-2
S3474-3
S3822-1
S4237-1
S4249-1
S3816-1
S3825-1
S3169-2
S3644-1
S3190-4
S3234-1
S3220-1
S3291-3
S4172-1
S4176-3
S4047-1
S4056-1
S3652-1
S3188-2
S3193-1
1
0.9
0.8
0.7
0.6
0.5
0.4
date_of
20180228
20180301
20180323
tumor_percentage_high
high
low
er
Negative
Positive
response
non-pCR
pCR
treatment
post
pre

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)
```


study_id
date_of
tumor_percentage_high
er
response
treatment
S3188-2
S3193-1
S4237-1
S4249-1
S3154-2
S3816-1
S3474-3
S3822-1
S3713-1
S3723-1
S4172-1
S4176-3
S3372-1
S3732-1
S3169-2
S4047-1
S3291-3
S3825-1
S3292-3
S3582-4
S3477-1
S4136-1
S3220-1
S3837-2
S3644-1
S3652-1
S3404-1
S3424-1
S3194-3
S4056-1
S3563-3
S3688-2
S4089-1
S4101-3
S3190-4
S3234-1
S3715-2
S4261-1
S4295-5
S3374-2
S3697-3
S4144-2
S3728-3
S3741-3
1
0.99
0.98
0.97
0.96
0.95
0.94
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
treatment
post
pre

Treatment Post vs Pre - see Table3

Response pCR vs non-pCR - see Table4

ER : Positive vs Negative - Table 5

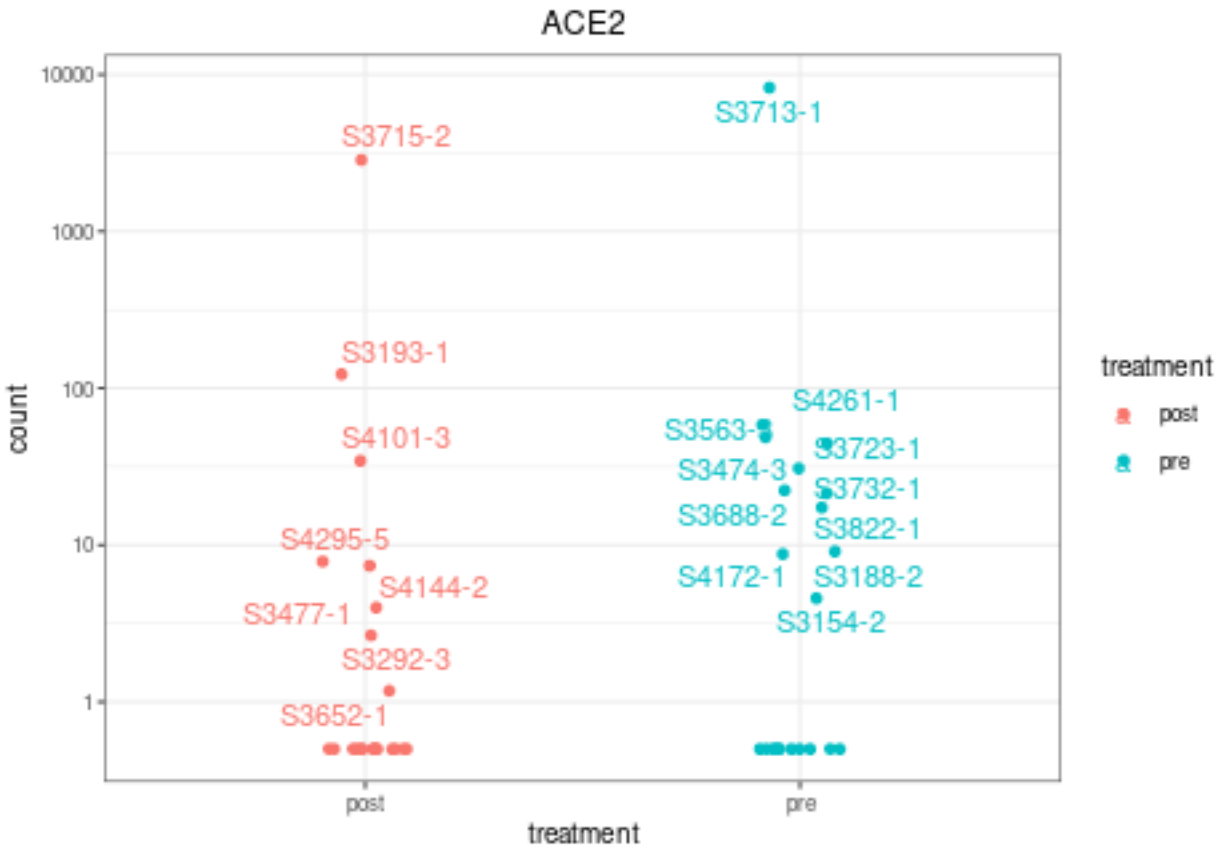
tumor_percentage_high : High vs Low - Table 6

date_of: 20180323 vs 20180228 - Table 7

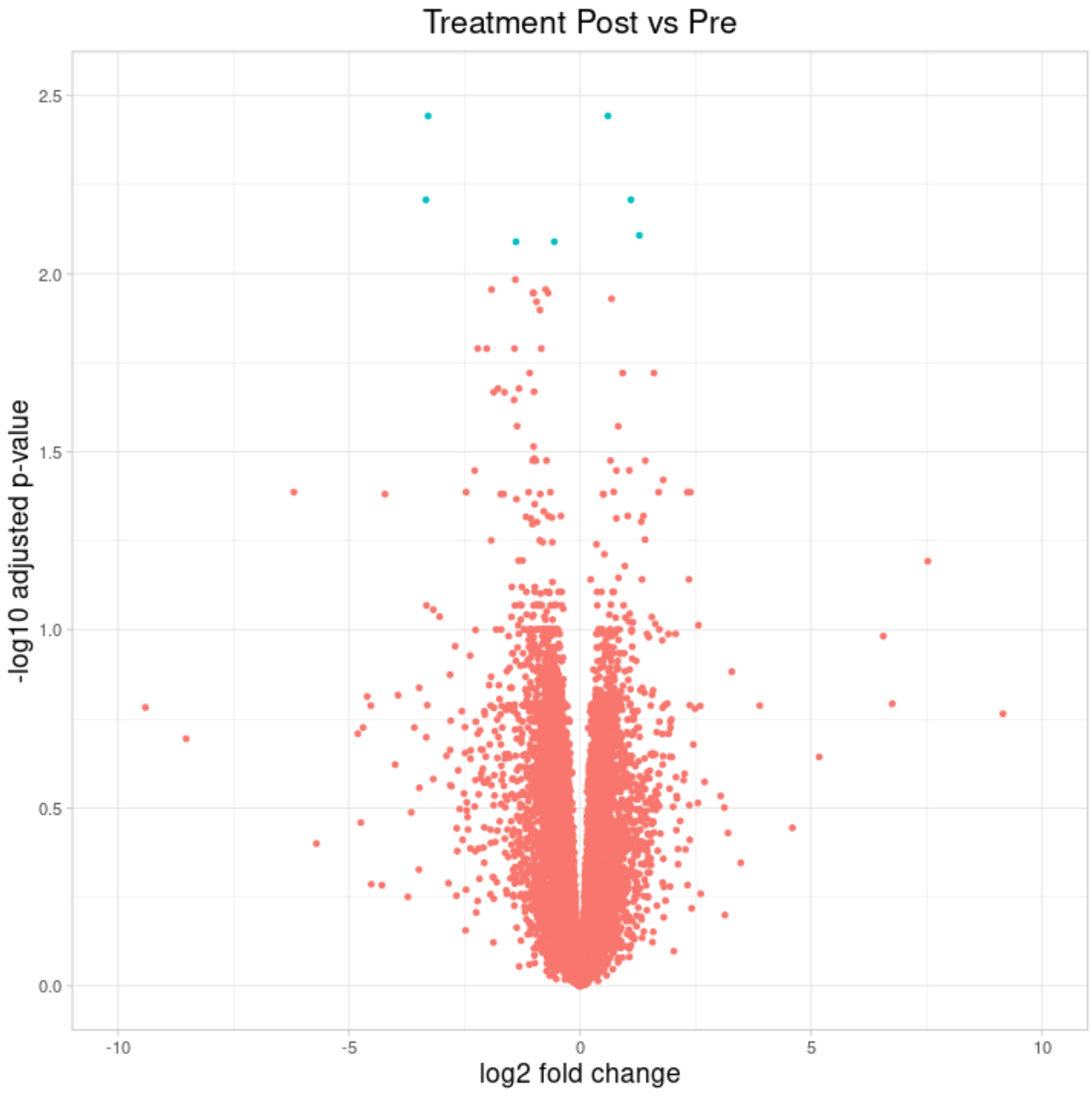
Visualization

Gene example

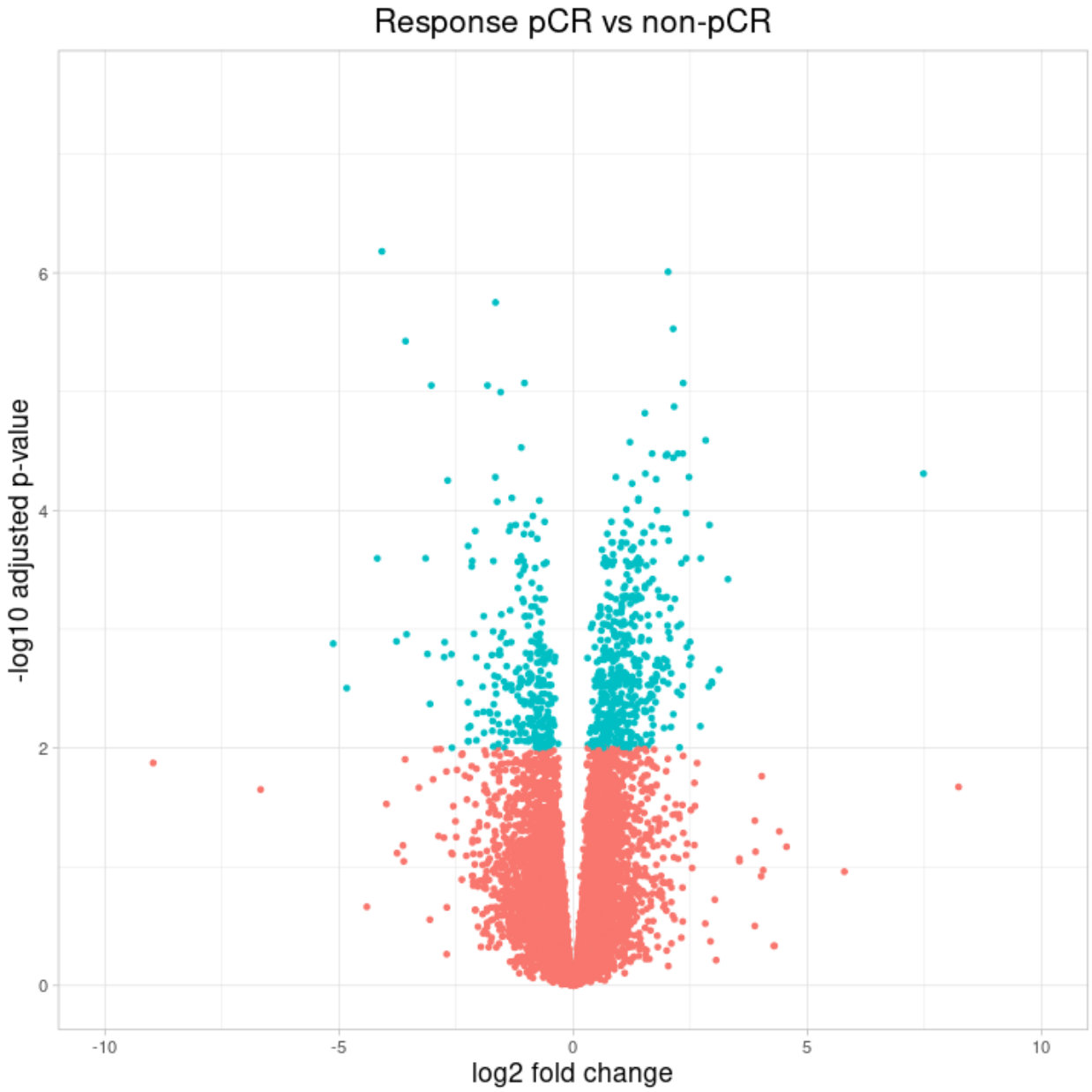
```
d <- plotCounts(dds,
                gene = "ENSG00000130234",
                intgroup = "treatment",
                returnData = TRUE)

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()
```



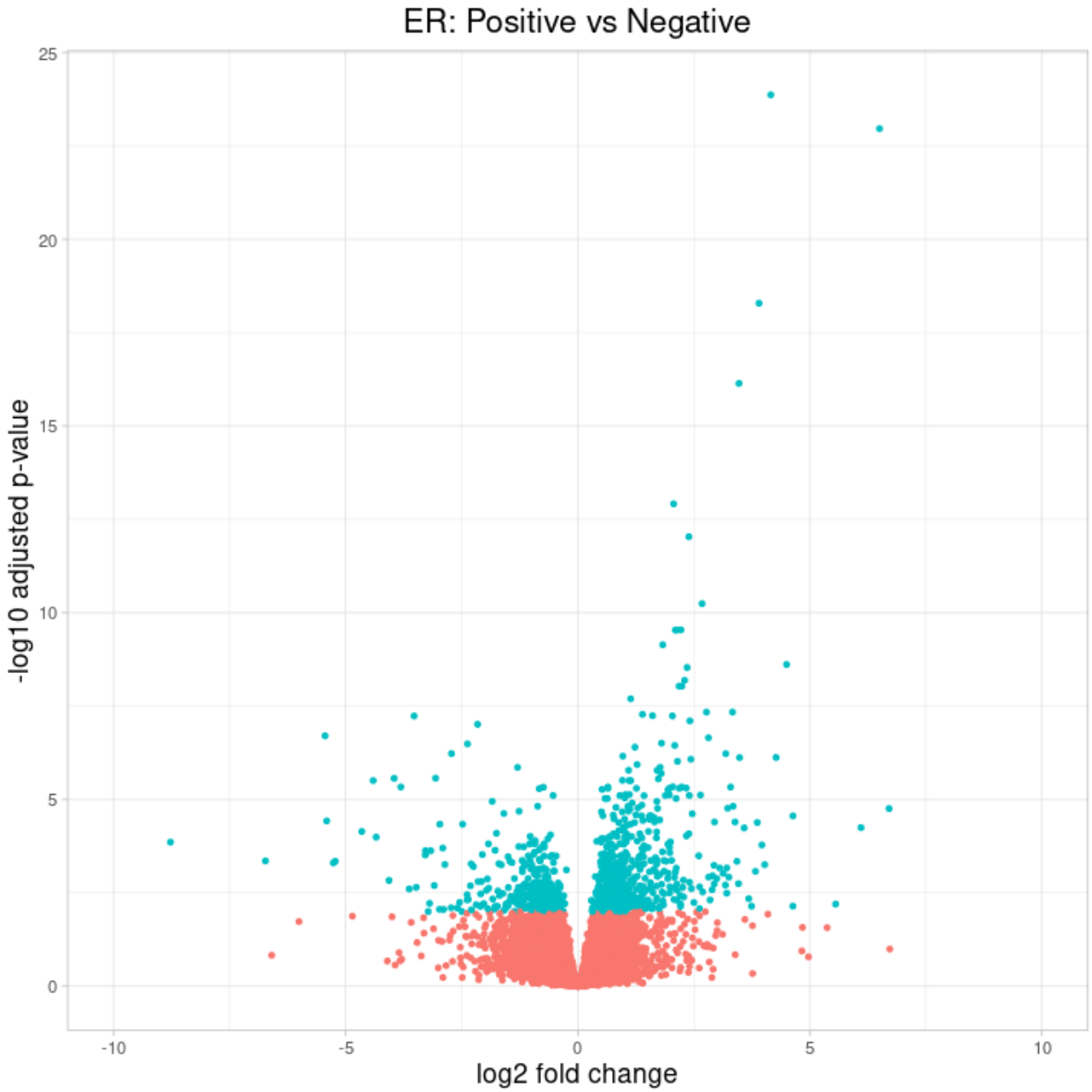
```r
# Add a column for significant genes
resTreatment_tb <- resTreatment_tb %>% mutate(threshold = padj < 0.01)

## 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)))
```



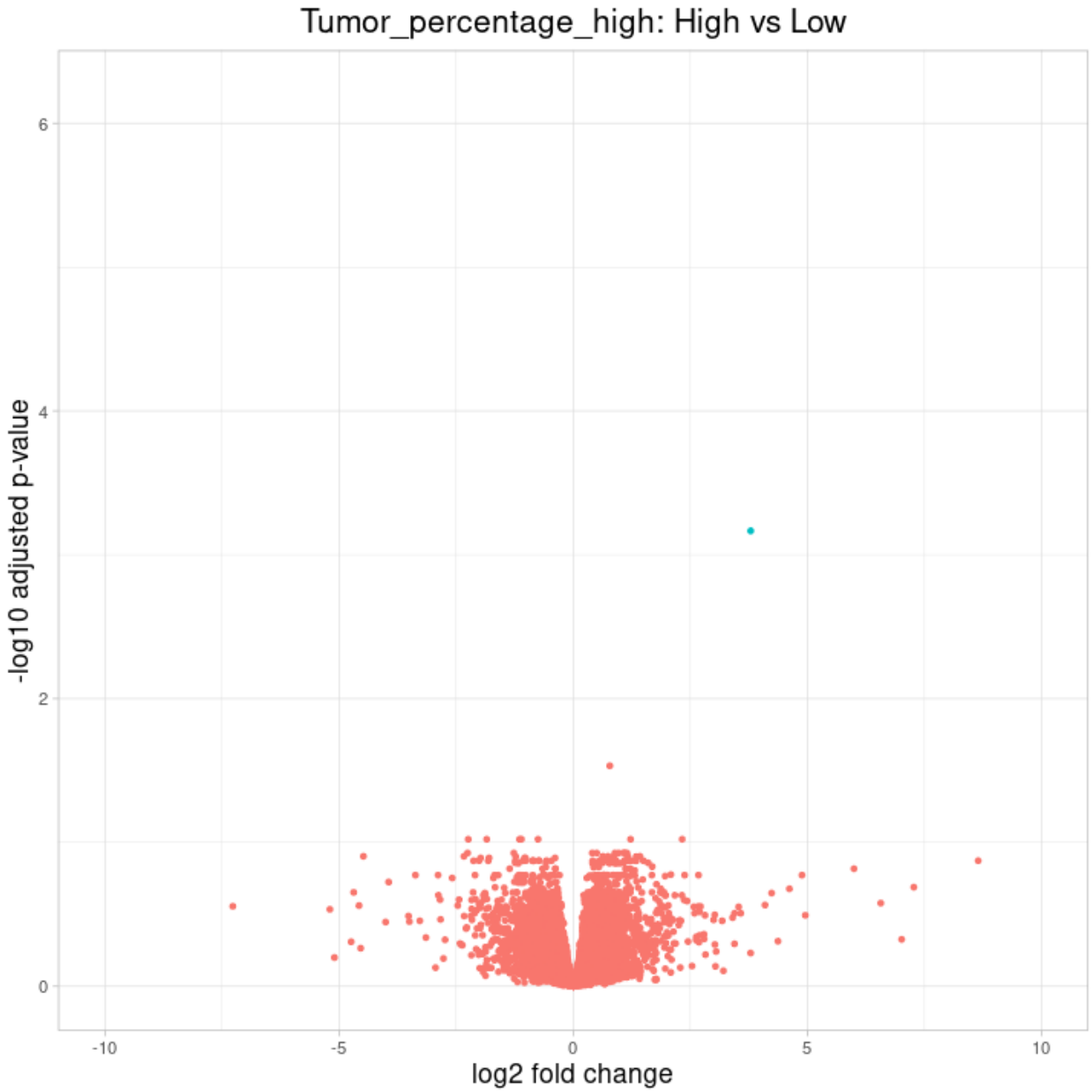
```r
# Add a column for significant genes
resResponse_tb <- resResponse_tb %>% mutate(threshold = padj < 0.01)

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)))
```



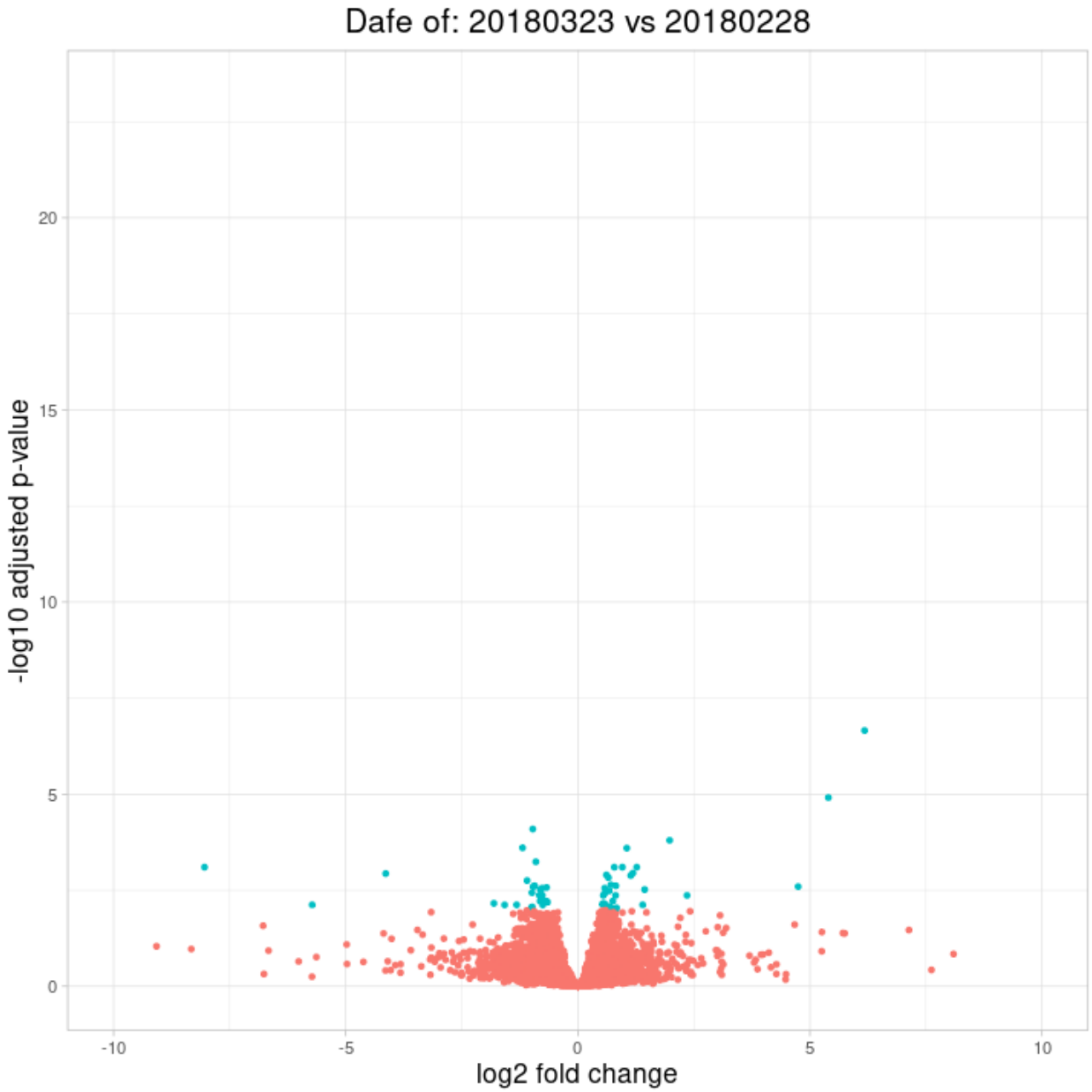
```r
# Add a column for significant genes
resER_tb <- resER_tb %>% mutate(threshold = padj < 0.01)

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)))
```

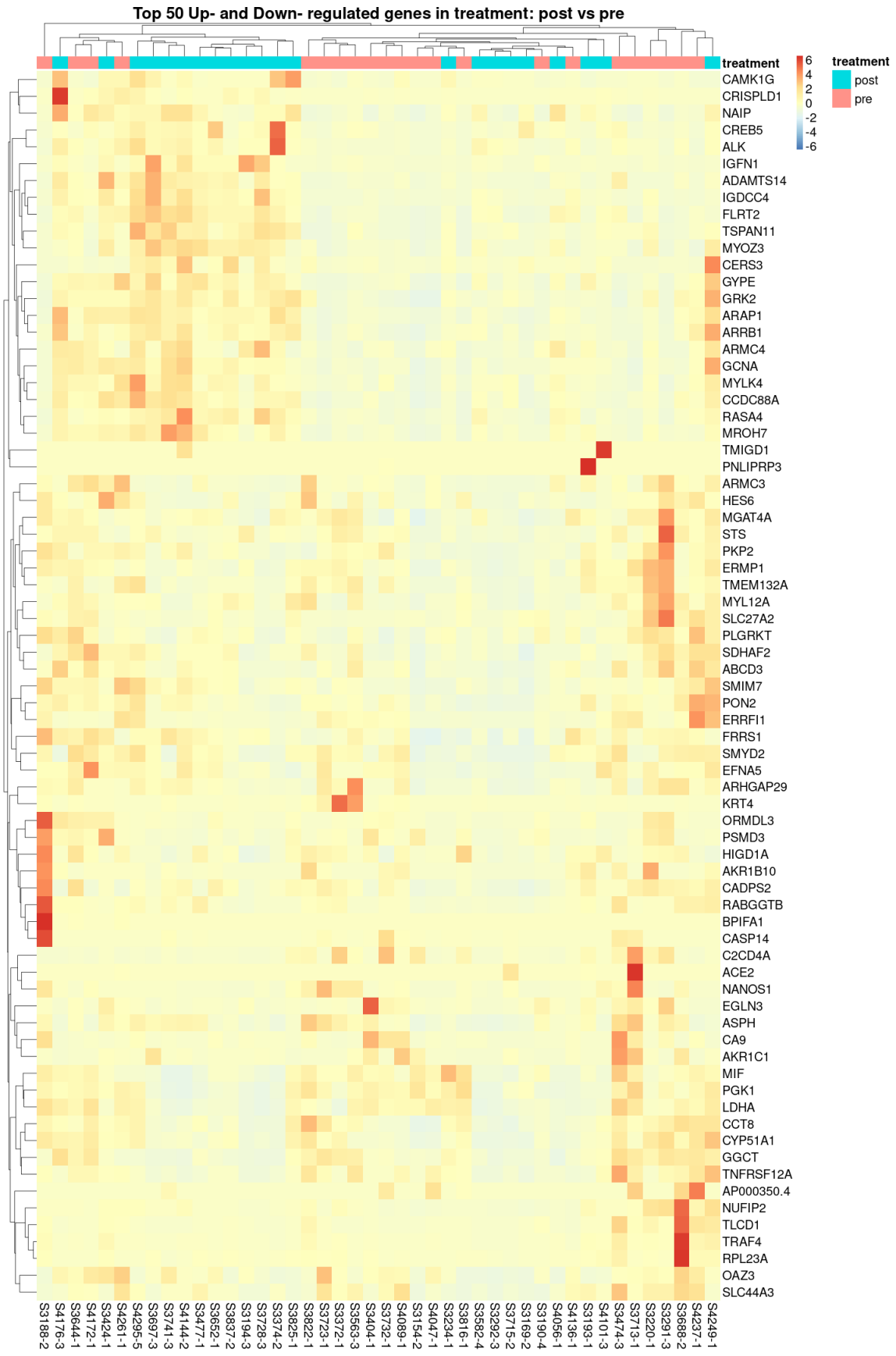


```r
# Add a column for significant genes
resDO_tb <- resDO_tb %>% mutate(threshold = padj < 0.01)

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)))
```

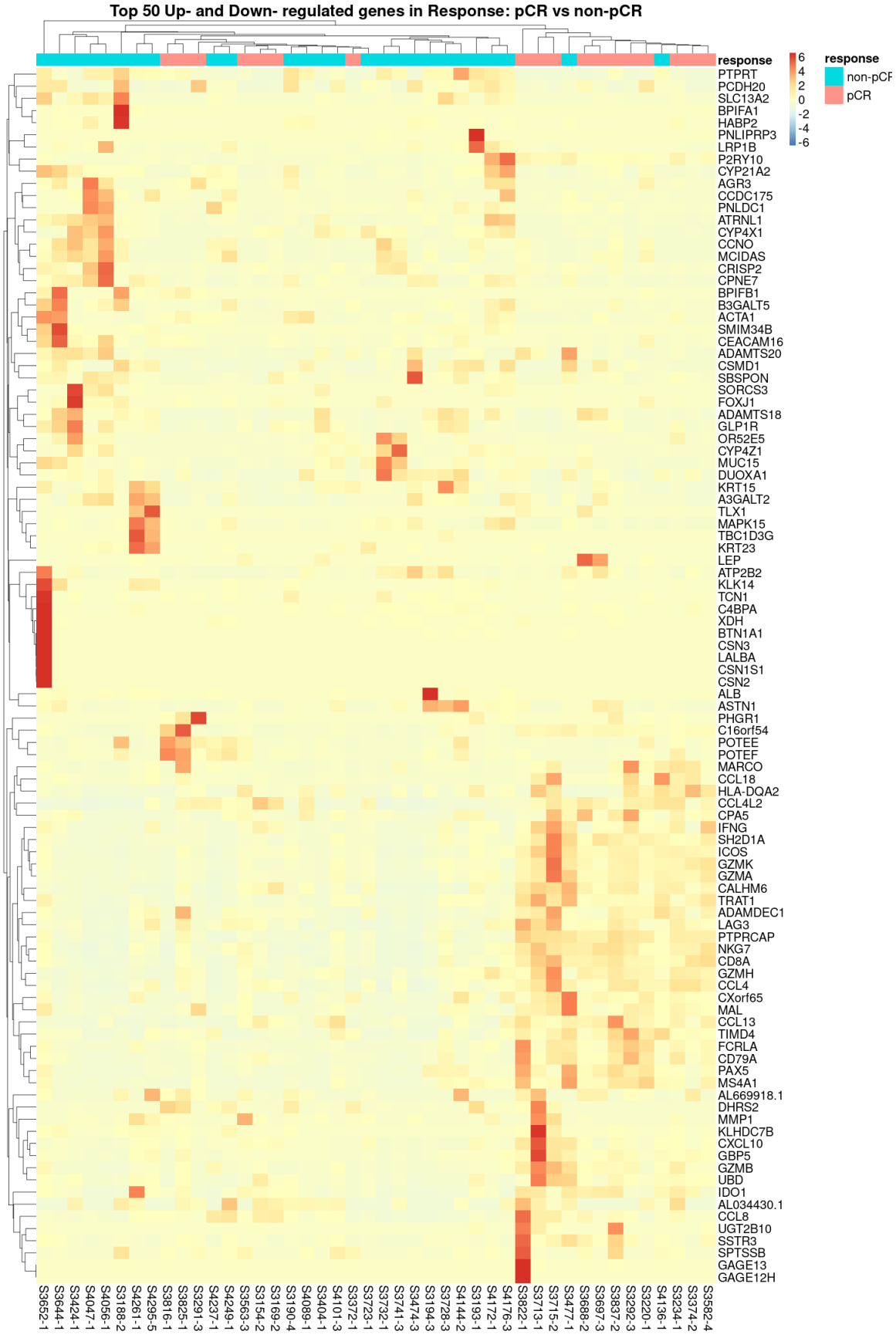


Heatmaps

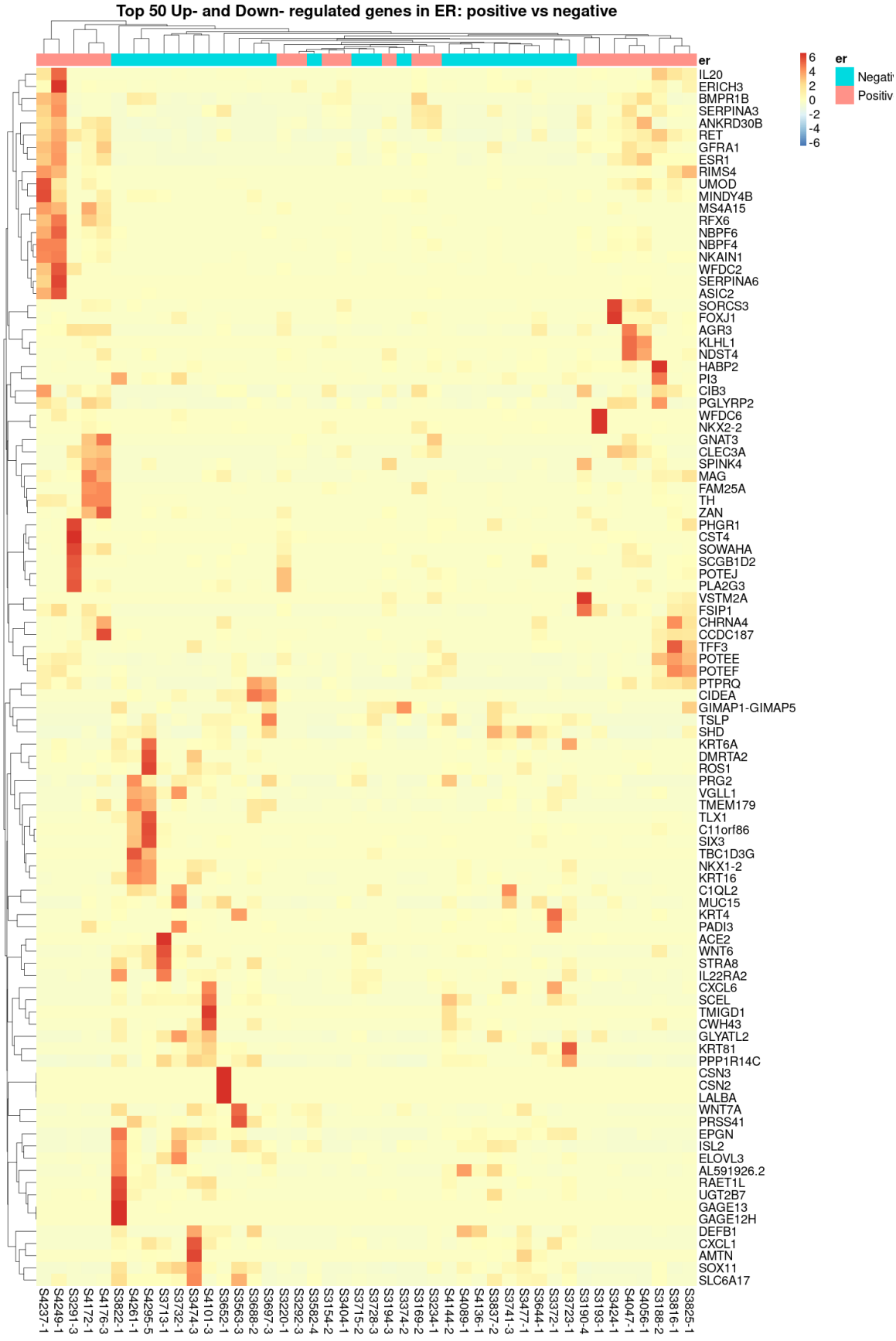
```
# Create a matrix of normalized expression
sig_up <- resTreatment_tb_significant %>% arrange(-log2FoldChange) %>% head(50) %>% pull(gene)
sig_down <- resTreatment_tb_significant %>% arrange(log2FoldChange) %>% head(50) %>% pull(gene)
sig <- c(sig_up, sig_down)

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)
```


Top 50 Up- and Down- regulated genes in treatment: post vs pre
treatment
post
pre
6
4
2
0
-2
-4
-6
CAMK1G
CRISPLD1
NAIP
CREB5
ALK
IGFN1
ADAMTS14
IGDCC4
FLRT2
TSPAN11
MYOZ3
CERS3
GYPE
GRK2
ARAP1
ARRB1
ARMC4
GCNA
MYLK4
CCDC88A
RASA4
MROH7
TMIGD1
PNLIPRP3
ARMC3
HES6
MGAT4A
STS
PKP2
ERMP1
TMEM132A
MYL12A
SLC27A2
PLGRKT
SDHAF2
ABCD3
SMIM7
PON2
ERRFI1
FRRS1
SMYD2
EFNA5
ARHGAP29
KRT4
ORMDL3
PSMD3
HIGD1A
AKR1B10
CADPS2
RABGGTB
BPIFA1
CASP14
C2CD4A
ACE2
NANOS1
EGLN3
ASPH
CA9
AKR1C1
MIF
PGK1
LDHA
CCT8
CYP51A1
GGCT
TNFRSF12A
AP000350.4
NUFIP2
TLCD1
TRAF4
RPL23A
OAZ3
SLC44A3
S3188-2
S4176-3
S3644-1
S4172-1
S3424-1
S4261-1
S4295-5
S3697-3
S3741-3
S4144-2
S3477-1
S3652-1
S3837-2
S3194-3
S3728-3
S3374-2
S3825-1
S3822-1
S3723-1
S3372-1
S3563-3
S3404-1
S3732-1
S4089-1
S3154-2
S4047-1
S3234-1
S3816-1
S3582-4
S3292-3
S3715-2
S3169-2
S3190-4
S4056-1
S4136-1
S3193-1
S4101-3
S3474-3
S3713-1
S3220-1
S3291-3
S3688-2
S4237-1
S4249-1

```r
# 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)
```

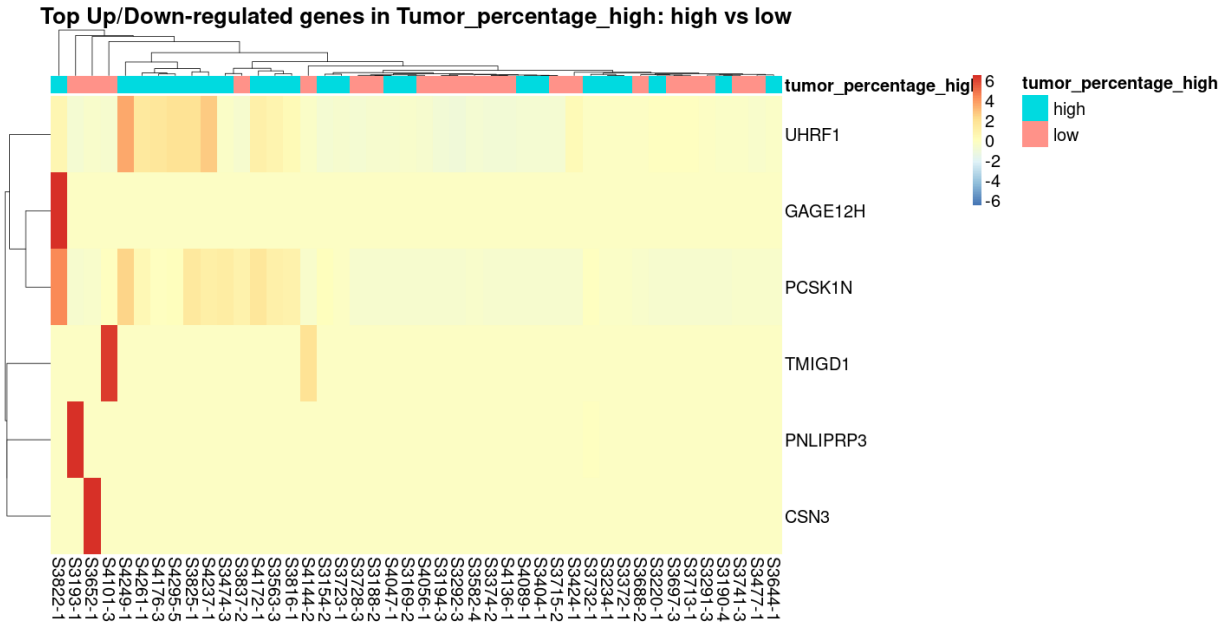

Top 50 Up- and Down- regulated genes in Response: pCR vs non-pCR
response
non-pCR
pCR
6
4
2
0
-2
-4
-6
PTPRT
PCDH20
SLC13A2
BPIFA1
HABP2
PNLIPRP3
LRP1B
P2RY10
CYP21A2
AGR3
CCDC175
PNLDC1
ATRNL1
CYP4X1
CCNO
MCIDAS
CRISP2
CPNE7
BPIFB1
B3GALT5
ACTA1
SMIM34B
CEACAM16
ADAMTS20
CSMD1
SBSPON
SORCS3
FOXJ1
ADAMTS18
GLP1R
OR52E5
CYP4Z1
MUC15
DUOXA1
KRT15
A3GALT2
TLX1
MAPK15
TBC1D3G
KRT23
LEP
ATP2B2
KLK14
TCN1
C4BPA
XDH
BTN1A1
CSN3
LALBA
CSN1S1
CSN2
ALB
ASTN1
PHGR1
C16orf54
POTEE
POTEF
MARCO
CCL18
HLA-DQA2
CCL4L2
CPA5
IFNG
SH2D1A
ICOS
GZMK
GZMA
CALHM6
TRAT1
ADAMDEC1
LAG3
PTPRCAP
NKG7
CD8A
GZMH
CCL4
CXorf65
MAL
CCL13
TIMD4
FCRLA
CD79A
PAX5
MS4A1
AL669918.1
DHRS2
MMP1
KLHDC7B
CXCL10
GBP5
GZMB
UBD
IDO1
AL034430.1
CCL8
UGT2B10
SSTR3
SPTSSB
GAGE13
GAGE12H
S3652-1
S3644-1
S3424-1
S4047-1
S4056-1
S3188-2
S4261-1
S4295-5
S3816-1
S3825-1
S3291-3
S4237-1
S4249-1
S3563-3
S3154-2
S3169-2
S3190-4
S4089-1
S3404-1
S4101-3
S3372-1
S3723-1
S3732-1
S3741-3
S3474-3
S3194-3
S3728-3
S4144-2
S3193-1
S4172-1
S4176-3
S3822-1
S3713-1
S3715-2
S3477-1
S3688-2
S3697-3
S3837-2
S3292-3
S3220-1
S4136-1
S3234-1
S3374-2
S3582-4

```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)
```

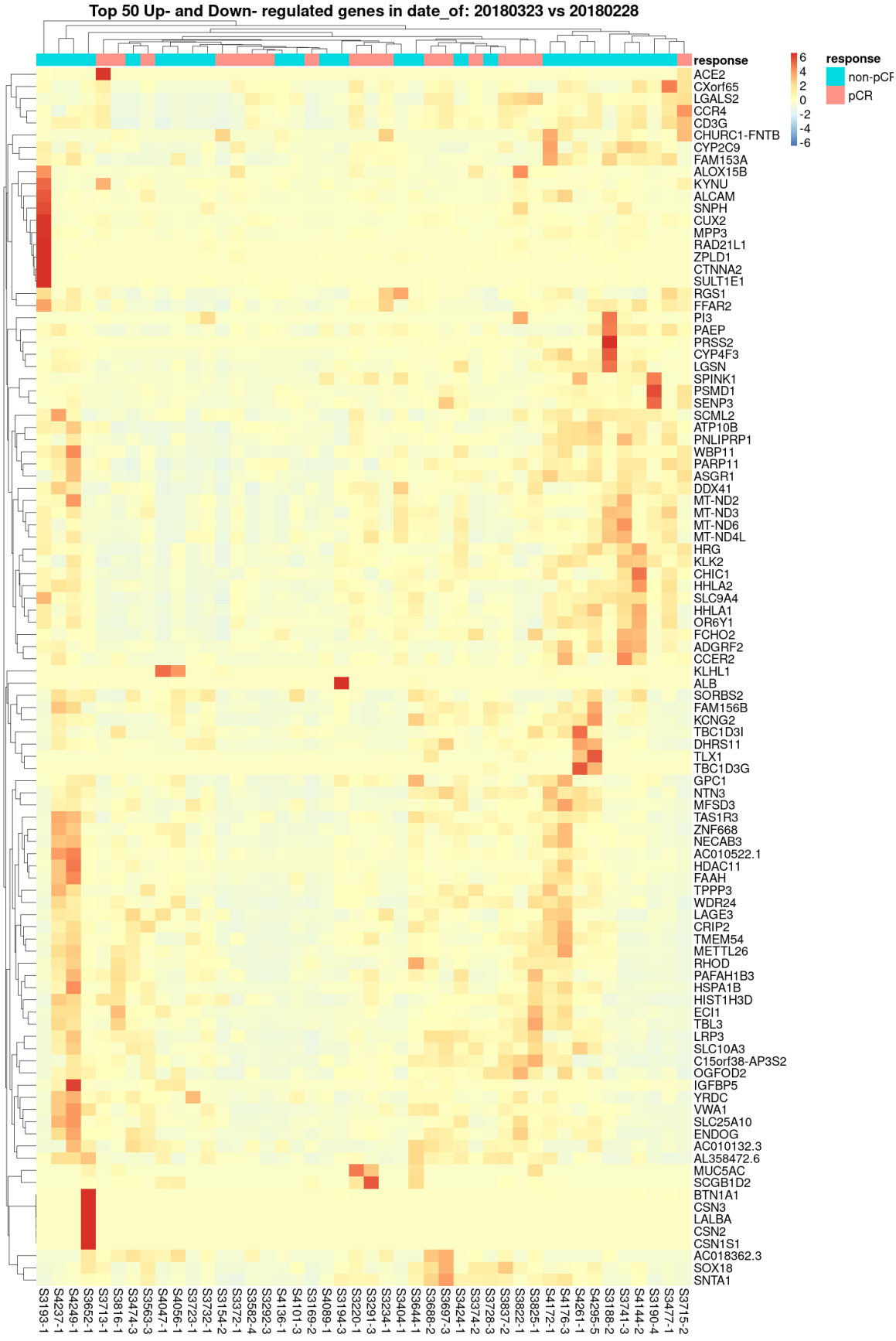

Top 50 Up- and Down- regulated genes in ER: positive vs negative
er
Negati
Positiv
6
4
2
0
-2
-4
-6
IL20
ERICH3
BMPR1B
SERPINA3
ANKRD30B
RET
GFRA1
ESR1
RIMS4
UMOD
MINDY4B
MS4A15
RFX6
NBPF6
NBPF4
NKAIN1
WFDC2
SERPINA6
ASIC2
SORCS3
FOXJ1
AGR3
KLHL1
NDST4
HABP2
PI3
CIB3
PGLYRP2
WFDC6
NKX2-2
GNAT3
CLEC3A
SPINK4
MAG
FAM25A
TH
ZAN
PHGR1
CST4
SOWAHA
SCGB1D2
POTEJ
PLA2G3
VSTM2A
FSIP1
CHRNA4
CCDC187
TFF3
POTEE
POTEF
PTPRQ
CIDEA
GIMAP1-GIMAP5
TSLP
SHD
KRT6A
DMRTA2
ROS1
PRG2
VGLL1
TMEM179
TLX1
C11orf86
SIX3
TBC1D3G
NKX1-2
KRT16
C1QL2
MUC15
KRT4
PADI3
ACE2
WNT6
STRA8
IL22RA2
CXCL6
SCEL
TMIGD1
CWH43
GLYATL2
KRT81
PPP1R14C
CSN3
CSN2
LALBA
WNT7A
PRSS41
EPGN
ISL2
ELOVL3
AL591926.2
RAET1L
UGT2B7
GAGE13
GAGE12H
DEFB1
CXCL1
AMTN
SOX11
SLC6A17
S4237-1
S4249-1
S3291-3
S4172-1
S4176-3
S3822-1
S4261-1
S4295-5
S3713-1
S3732-1
S3474-3
S4101-3
S3652-1
S3563-3
S3688-2
S3697-3
S3220-1
S3292-3
S3582-4
S3154-2
S3404-1
S3715-2
S3728-3
S3194-3
S3374-2
S3169-2
S3234-1
S4144-2
S4089-1
S4136-1
S3837-2
S3741-3
S3477-1
S3644-1
S3372-1
S3723-1
S3190-4
S3193-1
S3424-1
S4047-1
S4056-1
S3188-2
S3816-1
S3825-1

```r
# 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")
```

```r
# 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
ACE2
CXorf65
LGALS2
CCR4
CD3G
CHURC1-FNTB
CYP2C9
FAM153A
ALOX15B
KYNU
ALCAM
SNPH
CUX2
MPP3
RAD21L1
ZPLD1
CTNNA2
SULT1E1
RGS1
FFAR2
PI3
PAEP
PRSS2
CYP4F3
LGSN
SPINK1
PSMD1
SENP3
SCML2
ATP10B
PNLIPRP1
WBP11
PARP11
ASGR1
DDX41
MT-ND2
MT-ND3
MT-ND6
MT-ND4L
HRG
KLK2
CHIC1
HHLA2
SLC9A4
HHLA1
OR6Y1
FCHO2
ADGRF2
CCER2
KLHL1
ALB
SORBS2
FAM156B
KCNG2
TBC1D3I
DHRS11
TLX1
TBC1D3G
GPC1
NTN3
MFSD3
TAS1R3
ZNF668
NECAB3
AC010522.1
HDAC11
FAAH
TPPP3
WDR24
LAGE3
CRIP2
TMEM54
METTL26
RHOD
PAFAH1B3
HSPA1B
HIST1H3D
ECI1
TBL3
LRP3
SLC10A3
C15orf38-AP3S2
OGFOD2
IGFBP5
YRDC
VWA1
SLC25A10
ENDOG
AC010132.3
AL358472.6
MUC5AC
SCGB1D2
BTN1A1
CSN3
LALBA
CSN2
CSN1S1
AC018362.3
SOX18
SNTA1
S3193-1
S4237-1
S4249-1
S3652-1
S3713-1
S3816-1
S3474-3
S3563-3
S4047-1
S4056-1
S3723-1
S3732-1
S3154-2
S3372-1
S3582-4
S3292-3
S4136-1
S4101-3
S3169-2
S4089-1
S3194-3
S3220-1
S3291-3
S3234-1
S3404-1
S3644-1
S3688-2
S3697-3
S3424-1
S3374-2
S3728-3
S3837-2
S3822-1
S3825-1
S4172-1
S4176-3
S4261-1
S4295-5
S3188-2
S3741-3
S4144-2
S3190-4
S3477-1
S3715-2
6
4
2
0
-2
-4
-6
response
non-pCR
pCR

Prepare files for GSEA

GSEA day-wise

```
result_file <-  paste0("tables/all_samples.4gsea.day_wise.txt")

samples_post <- meta %>% dplyr::filter(treatment == "post") %>% row.names()
samples_pre <- meta %>% dplyr::filter(treatment == "pre") %>% row.names()

counts_gsea <- counts_gsea[,c("NAME", "DESCRIPTION", samples_post, samples_pre)]
# gsea now supports ENSEMBL_IDs
write_tsv(counts_gsea, result_file)


treatment_vector <- c(rep("post", length(samples_post)), rep("pre", length(samples_pre)))
treatment_str <- paste(treatment_vector, collapse = " ")

file_conn <- file("tables/all_samples.4gsea.day_wise.cls")
writeLines(c("44 2 1",
             "# post pre",
             treatment_str),
             file_conn)
close(file_conn)
```

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] ggplotify_0.0.6             ensembldb_2.14.1
##  [3] AnnotationFilter_1.14.0     GenomicFeatures_1.42.3
##  [5] AnnotationDbi_1.52.0        AnnotationHub_2.22.1
##  [7] BiocFileCache_1.14.0        dbplyr_2.1.1
```

```
##  [9] knitr_1.30                  ggrepel_0.9.1
## [11] tximport_1.18.0             DEGreport_1.26.0
## [13] pheatmap_1.0.12             RColorBrewer_1.1-2
## [15] forcats_0.5.1               stringr_1.4.0
## [17] dplyr_1.0.5                 purrr_0.3.4
## [19] readr_1.4.0                 tidyr_1.1.3
## [21] tibble_3.1.1                ggplot2_3.3.3
## [23] tidyverse_1.3.1             DESeq2_1.30.1
## [25] SummarizedExperiment_1.20.0 Biobase_2.50.0
## [27] MatrixGenerics_1.2.1        matrixStats_0.58.0
## [29] GenomicRanges_1.42.0        GenomeInfoDb_1.26.7
## [31] IRanges_2.24.1              S4Vectors_0.28.1
## [33] 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                  gridGraphics_0.5-1
##  [53] bit_4.0.4                    httr_1.4.2
##  [55] ellipsis_0.3.1               farver_2.1.0
##  [57] pkgconfig_2.0.3              reshape_0.8.8
##  [59] XML_3.99-0.6                 locfit_1.5-9.4
##  [61] utf8_1.2.1                   labeling_0.4.2
##  [63] tidyselect_1.1.0             rlang_0.4.10
##  [65] later_1.2.0                  munsell_0.5.0
##  [67] BiocVersion_3.12.0           cellranger_1.1.0
##  [69] tools_4.0.3                  cachem_1.0.4
##  [71] cli_2.5.0                    generics_0.1.0
##  [73] RSQLite_2.2.7                broom_0.7.6
##  [75] evaluate_0.14                fastmap_1.1.0
##  [77] ggdendro_0.1.22              yaml_2.2.1
```

```
##  [79] bit64_4.0.5                 fs_1.5.0
##  [81] nlme_3.1-149                quantreg_5.85
##  [83] mime_0.9                    xml2_1.3.2
##  [85] biomaRt_2.46.3              compiler_4.0.3
##  [87] rstudioapi_0.13             curl_4.3
##  [89] png_0.1-7                   interactiveDisplayBase_1.28.0
##  [91] reprex_2.0.0                geneplotter_1.68.0
##  [93] stringi_1.5.3               lattice_0.20-41
##  [95] ProtGenerics_1.22.0         Matrix_1.2-18
##  [97] psych_2.1.3                 vctrs_0.3.7
##  [99] pillar_1.6.0                lifecycle_1.0.0
## [101] BiocManager_1.30.12         GlobalOptions_0.1.2
## [103] conquer_1.0.2               cowplot_1.1.1
## [105] bitops_1.0-7                rtracklayer_1.50.0
## [107] httpuv_1.6.0                R6_2.5.0
## [109] promises_1.2.0.1            MASS_7.3-53
## [111] assertthat_0.2.1            openssl_1.4.3
## [113] rjson_0.2.20                withr_2.4.2
## [115] GenomicAlignments_1.26.0    Rsamtools_2.6.0
## [117] mnormt_2.0.2                GenomeInfoDbData_1.2.4
## [119] hms_1.0.0                   grid_4.0.3
## [121] rvcheck_0.1.8               rmarkdown_2.5
## [123] Cairo_1.5-12.2              logging_0.10-108
## [125] shiny_1.6.0                 lubridate_1.7.10
```