DE analysis - Day8

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 Day8 samples

```r
# Read in  a tx2gene file with transcript identifiers in the first column and gene identifiers in the s
tx2gene <- hsdb[, c("tx_id", "gene_id")]


# Run tximport
txi_file <- "data/txi.day8.RDS"
if (!rebuild_rds & file.exists(txi_file)){
    txi <- readRDS(txi_file)
}else{
    files <- files[rownames(meta)]
    txi <- tximport(files,
                type = "salmon",
                tx2gene = tx2gene,
                countsFromAbundance = "lengthScaledTPM",
                ignoreTxVersion = FALSE)
    saveRDS(txi, txi_file)
}

# Look at the counts
class(txi)
```

```
## [1] "list"
```

```r
attributes(txi)
```

```
## $names
## [1] "abundance"          "counts"                "length"
## [4] "countsFromAbundance"
```

2

```
txi$counts %>% View()
```

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    20
```

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
```

```
# Not the same? Make them the same
### This will change depending on what names you have listed!
#paste0(meta$samplename, "_", meta$library)
#rownames(meta) <- paste0(meta$samplename, "_", meta$library)
#meta$genotype <- relevel(meta$genotype, ref="Wildtype")

### 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.

```r
## Create DESeq2Dataset object
dds_file <- "data/dds.day8.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

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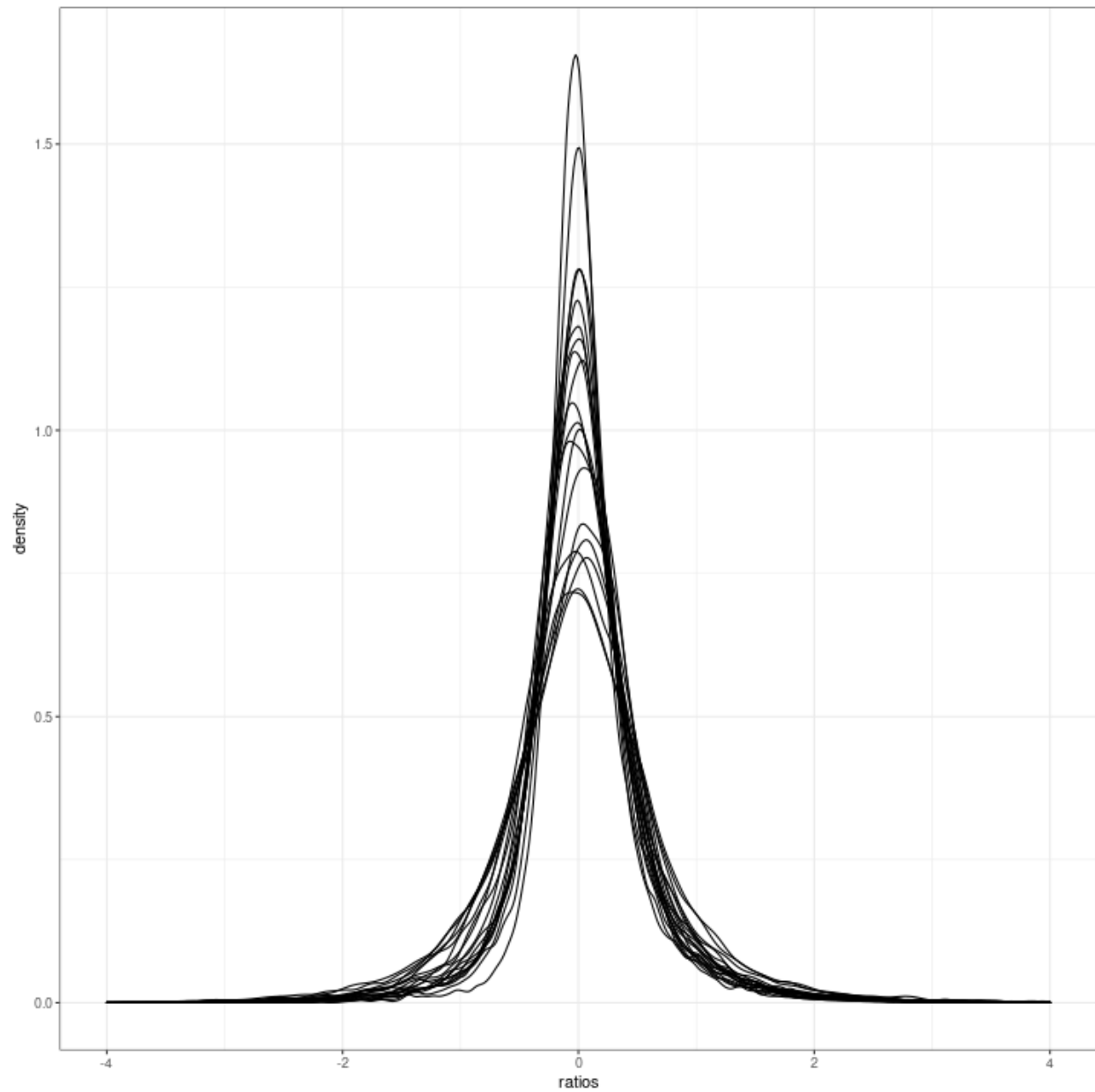
Mean-Variance QC plots

response

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

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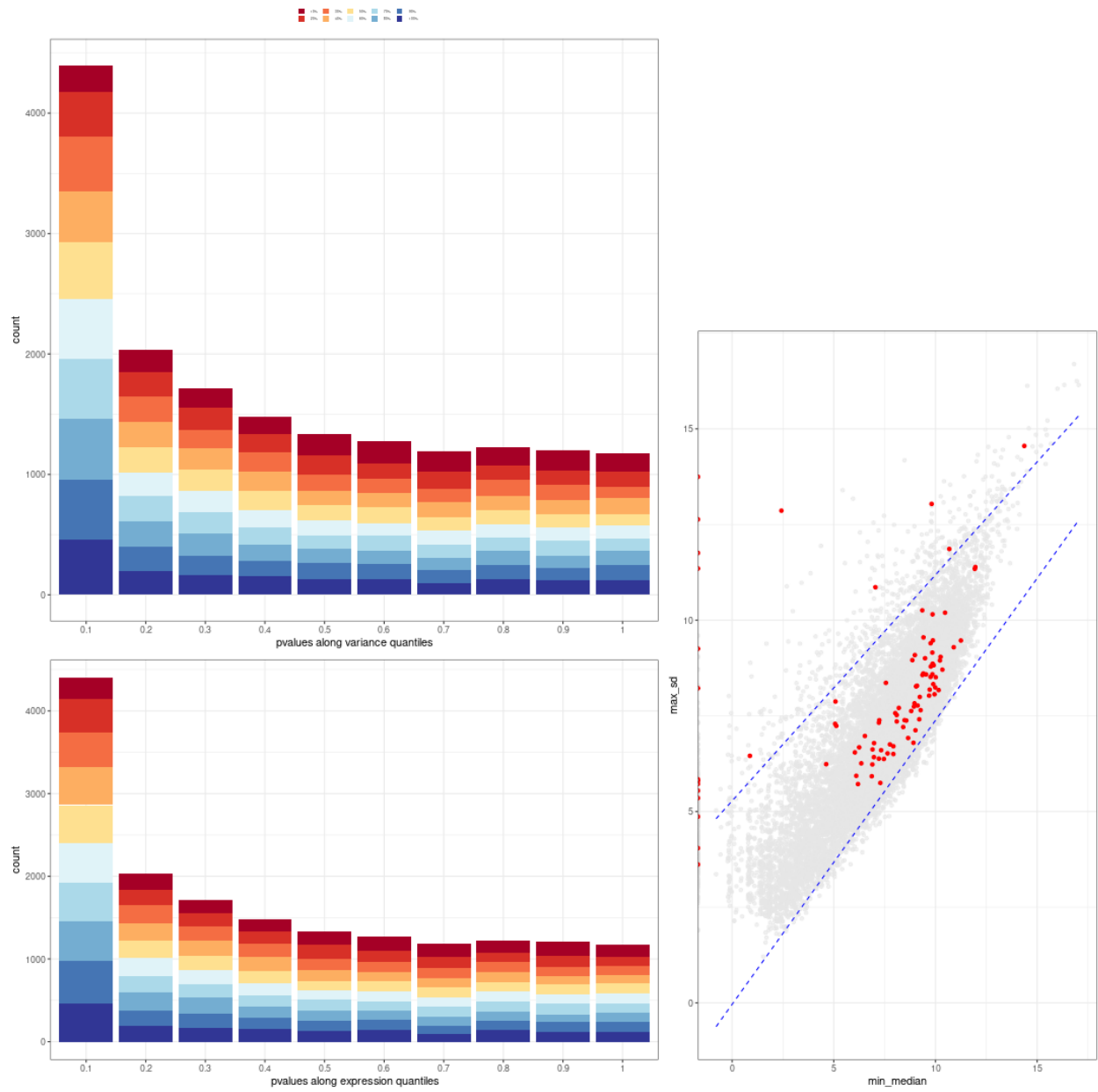
```r
degQC(counts, design[["er"]], pvalue = res[["pvalue"]])
```
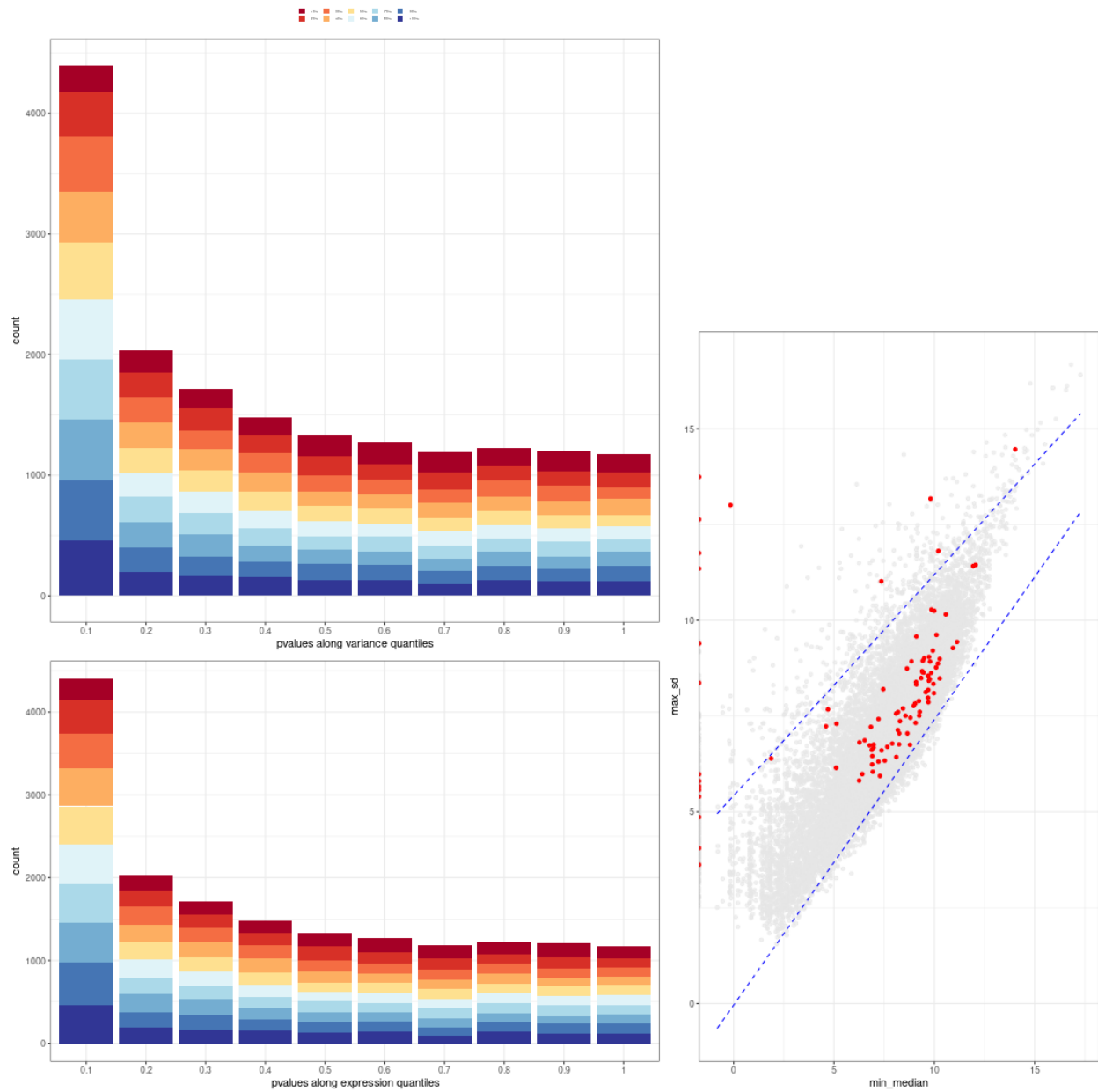
tumor__percentage__high

```
degQC(counts, design[["tumor_percentage_high"]], pvalue = res[["pvalue"]])
```
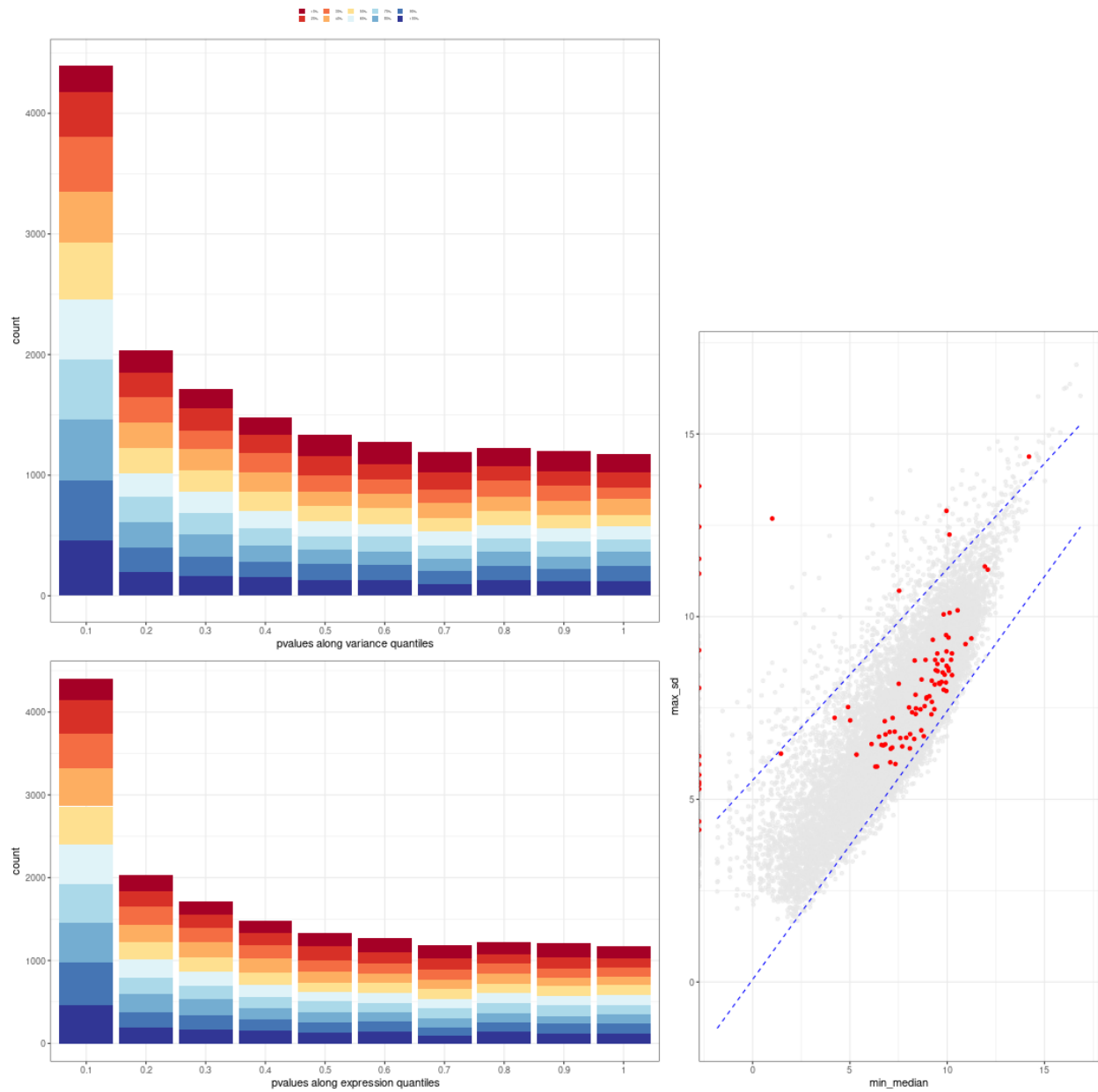
Covariates effect on count data

```
mdata <- colData(dds) %>% as.data.frame() %>%
  dplyr::select(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")
```
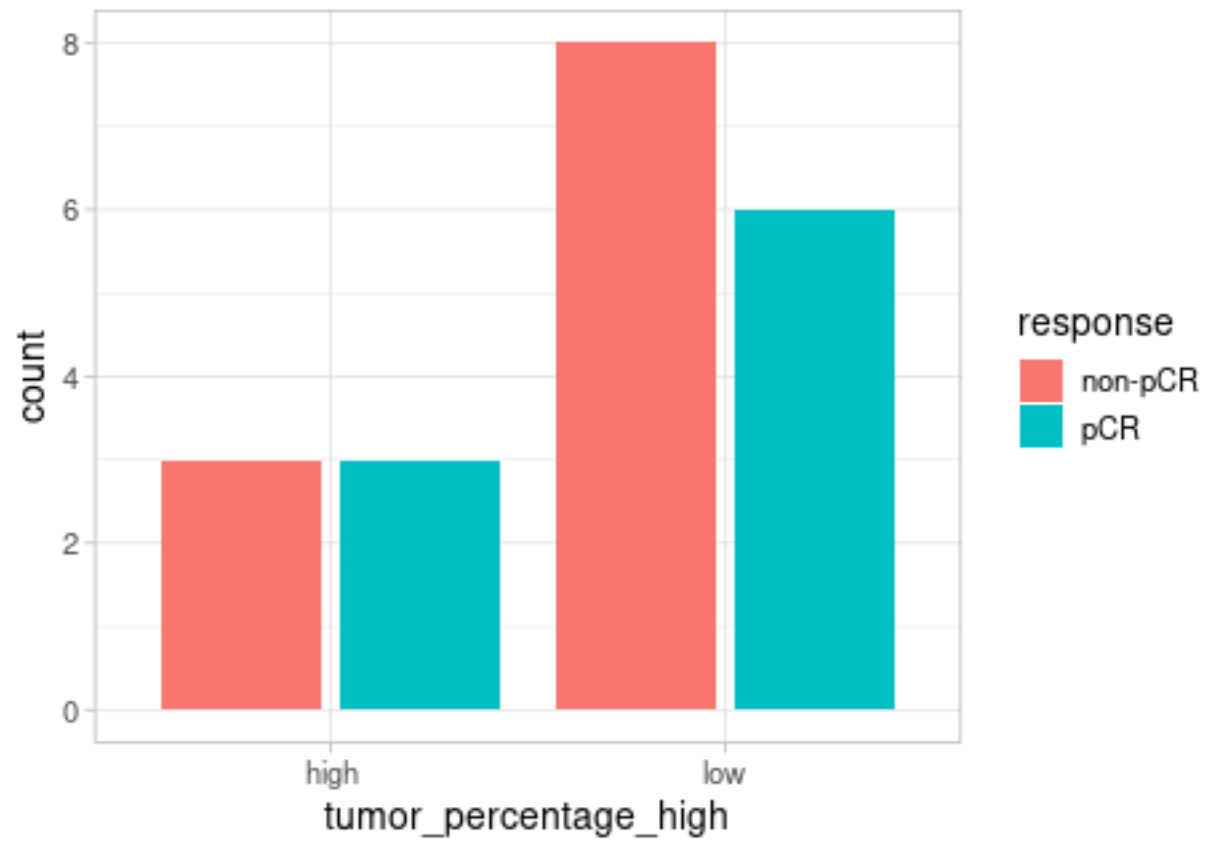
Covariates correlation with metrics

```
cor <- degCorCov(mdata)
```
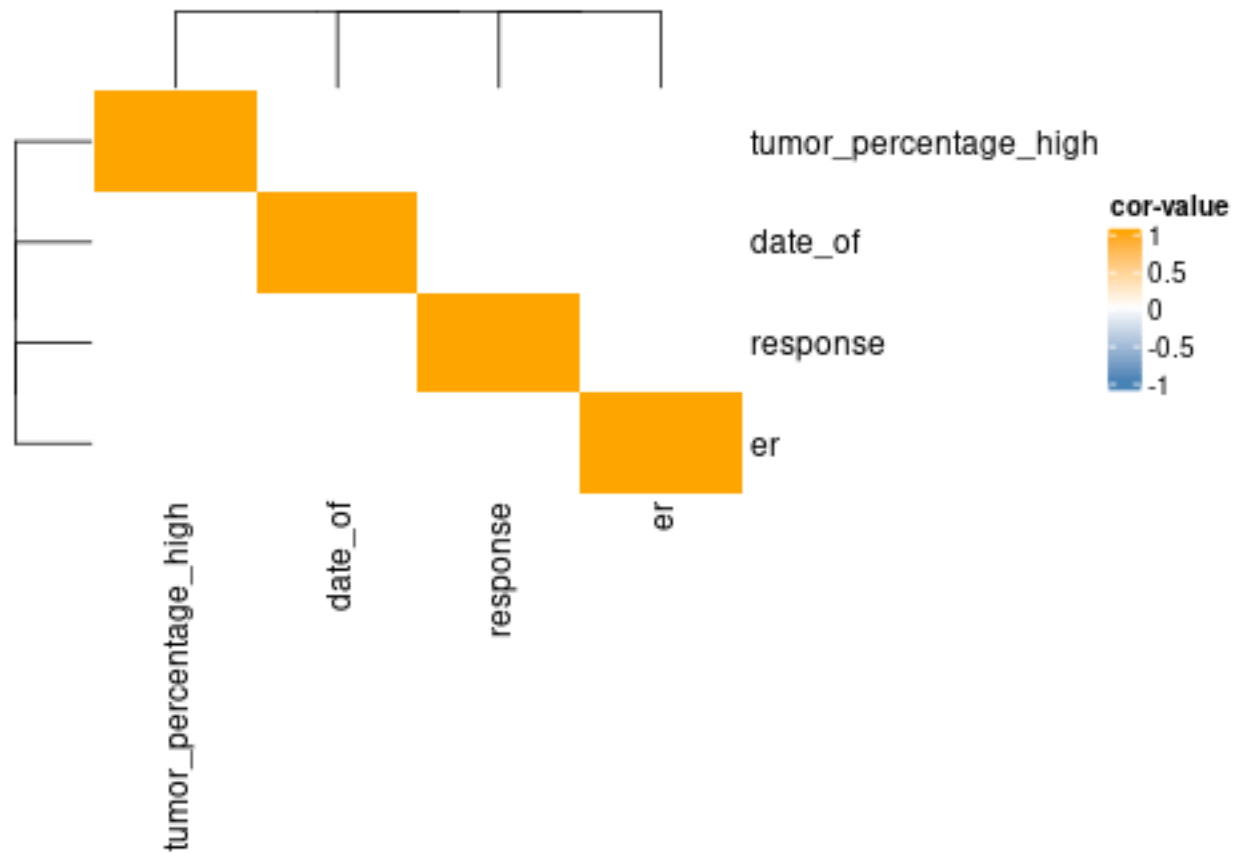
```r
mdata %>% ggplot(aes(date_of, fill = response)) + geom_bar(position = "dodge2")
```
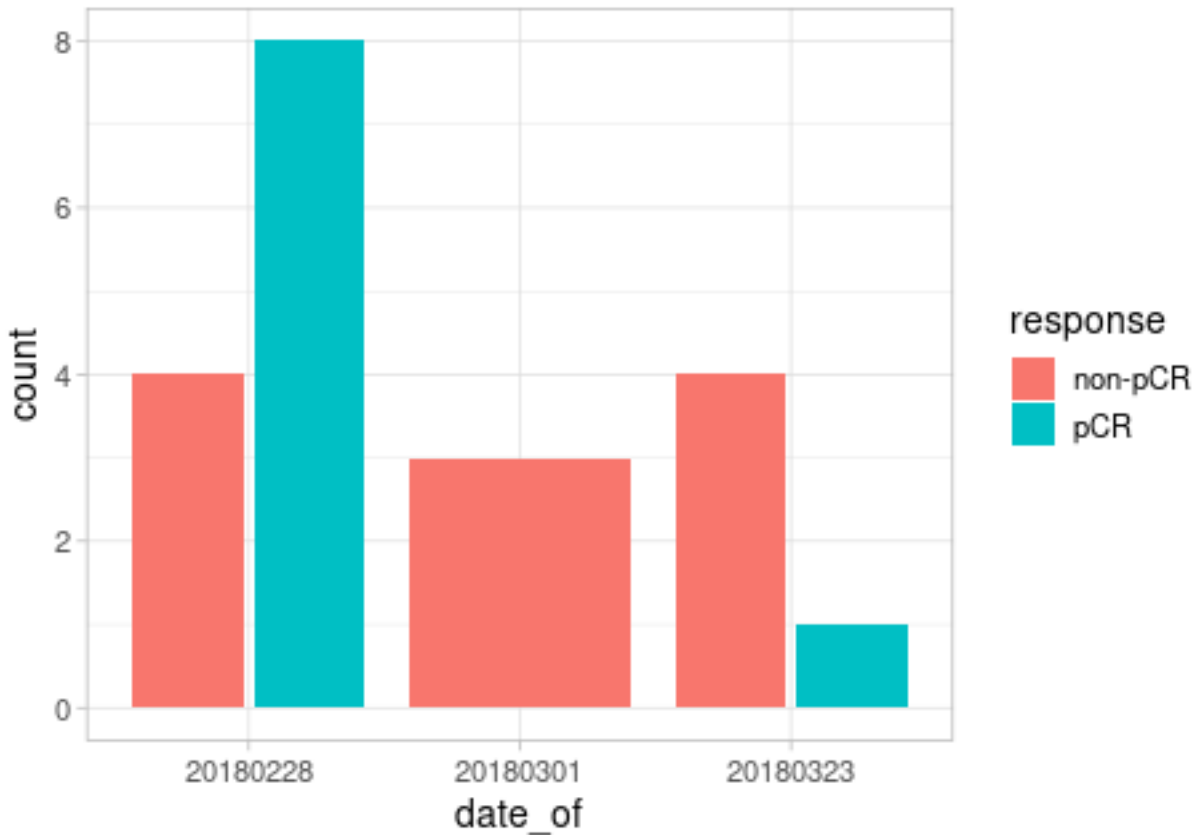
Sample-level QC analysis

```
### Transform counts for data visualization (unsupervised analysis)
rld_file <- "data/rld.day8.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

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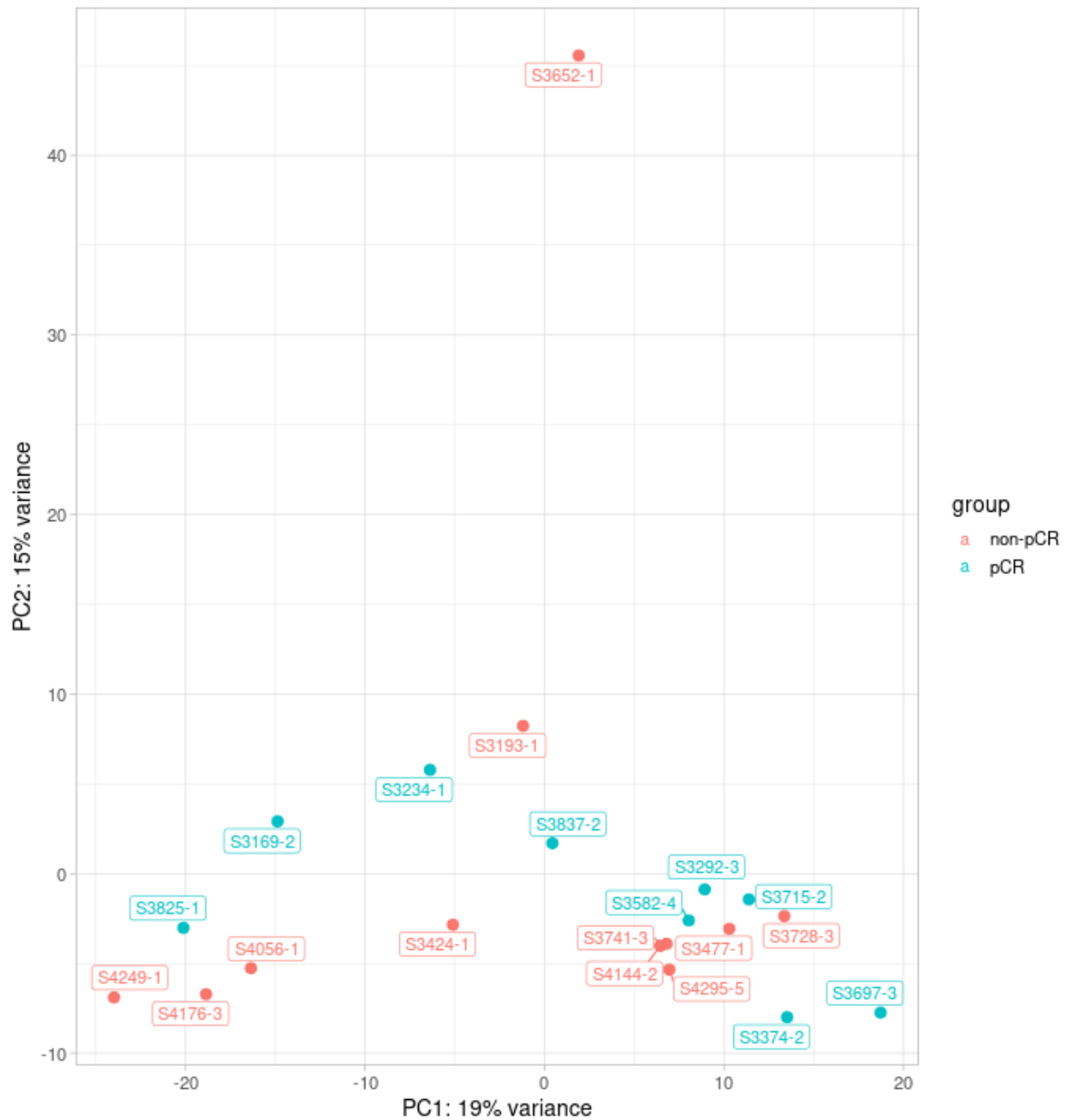
PCA - ER

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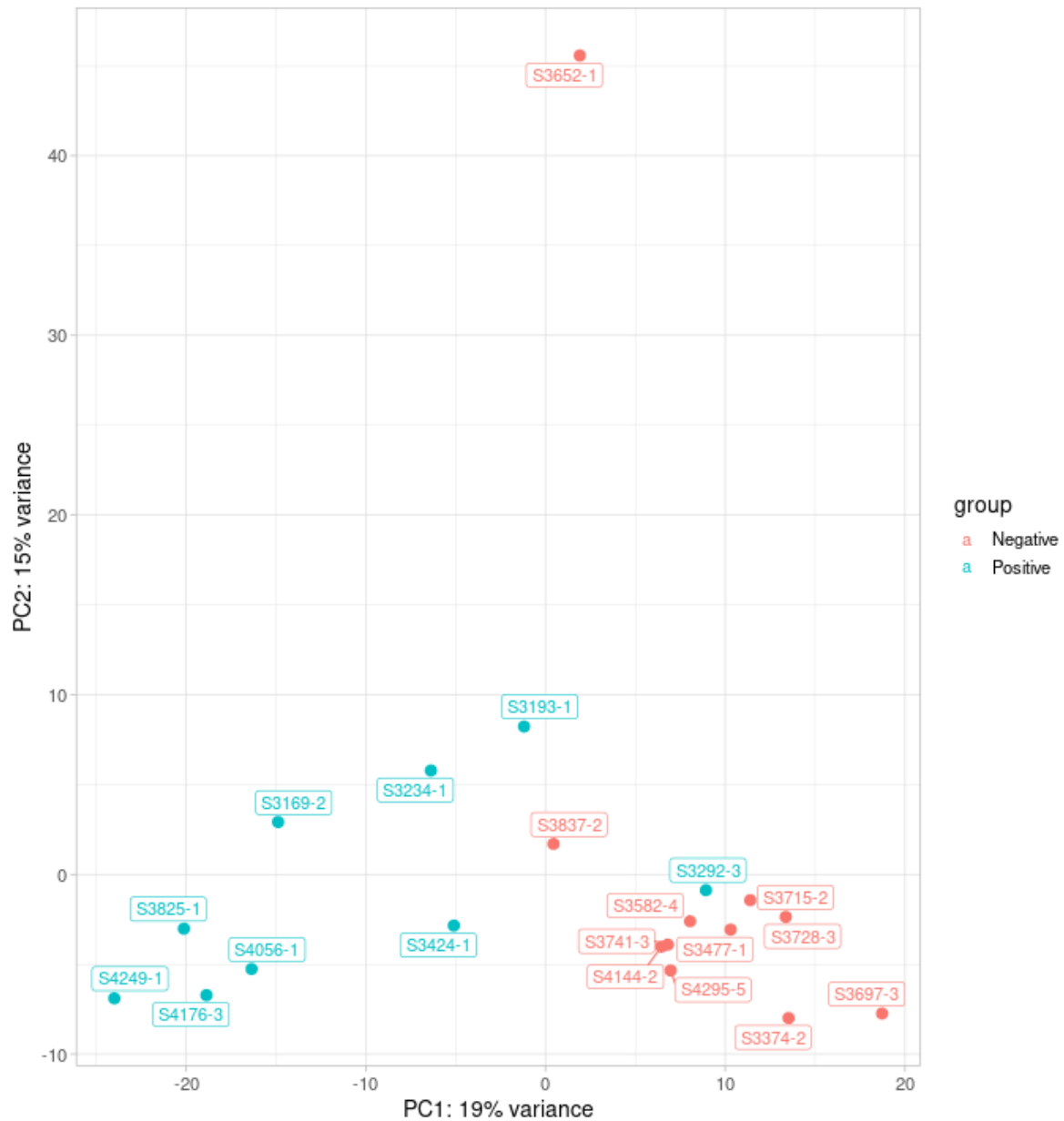
PCA - tumor_percentage

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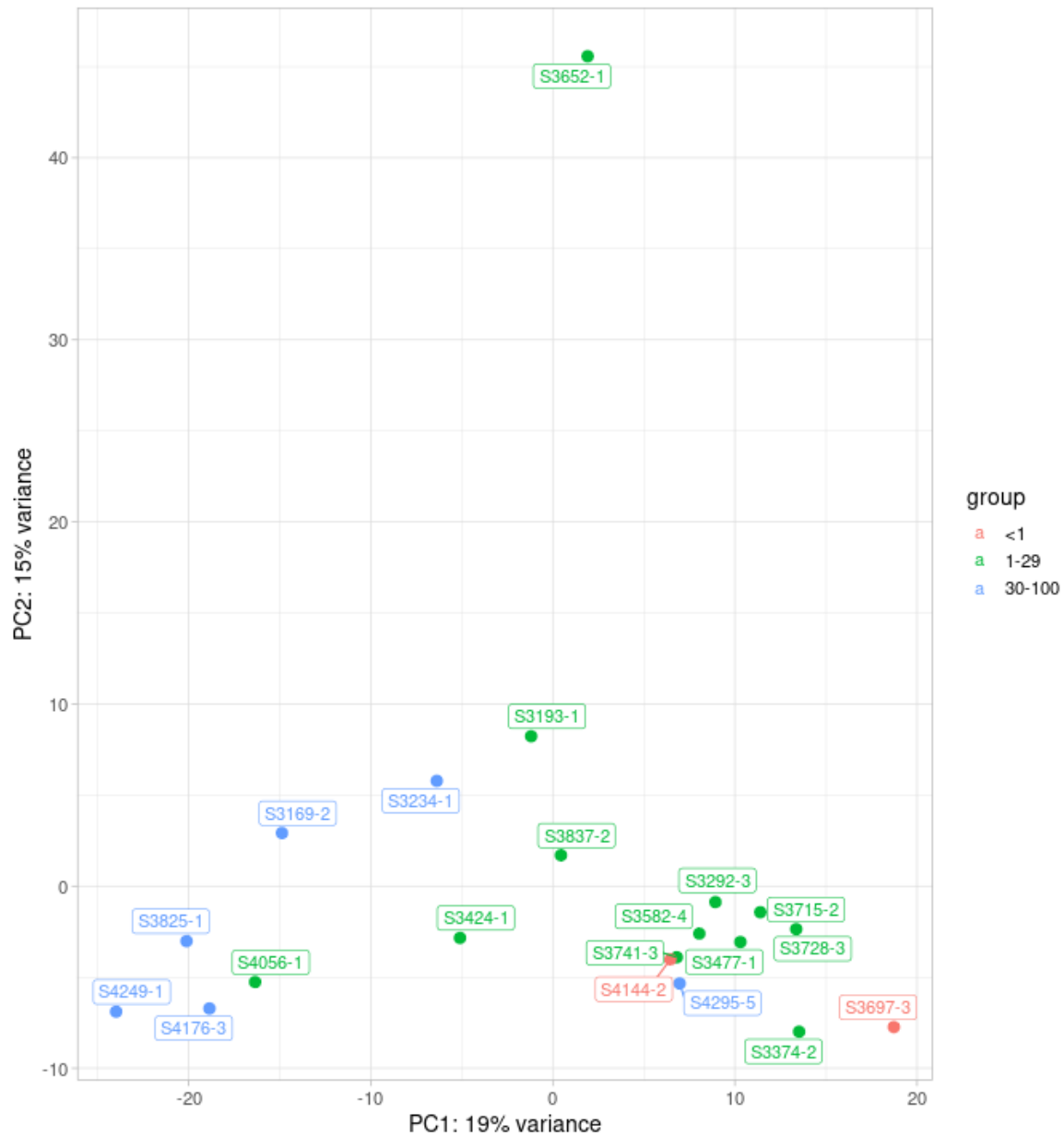
PCA - tumor_percentage_high

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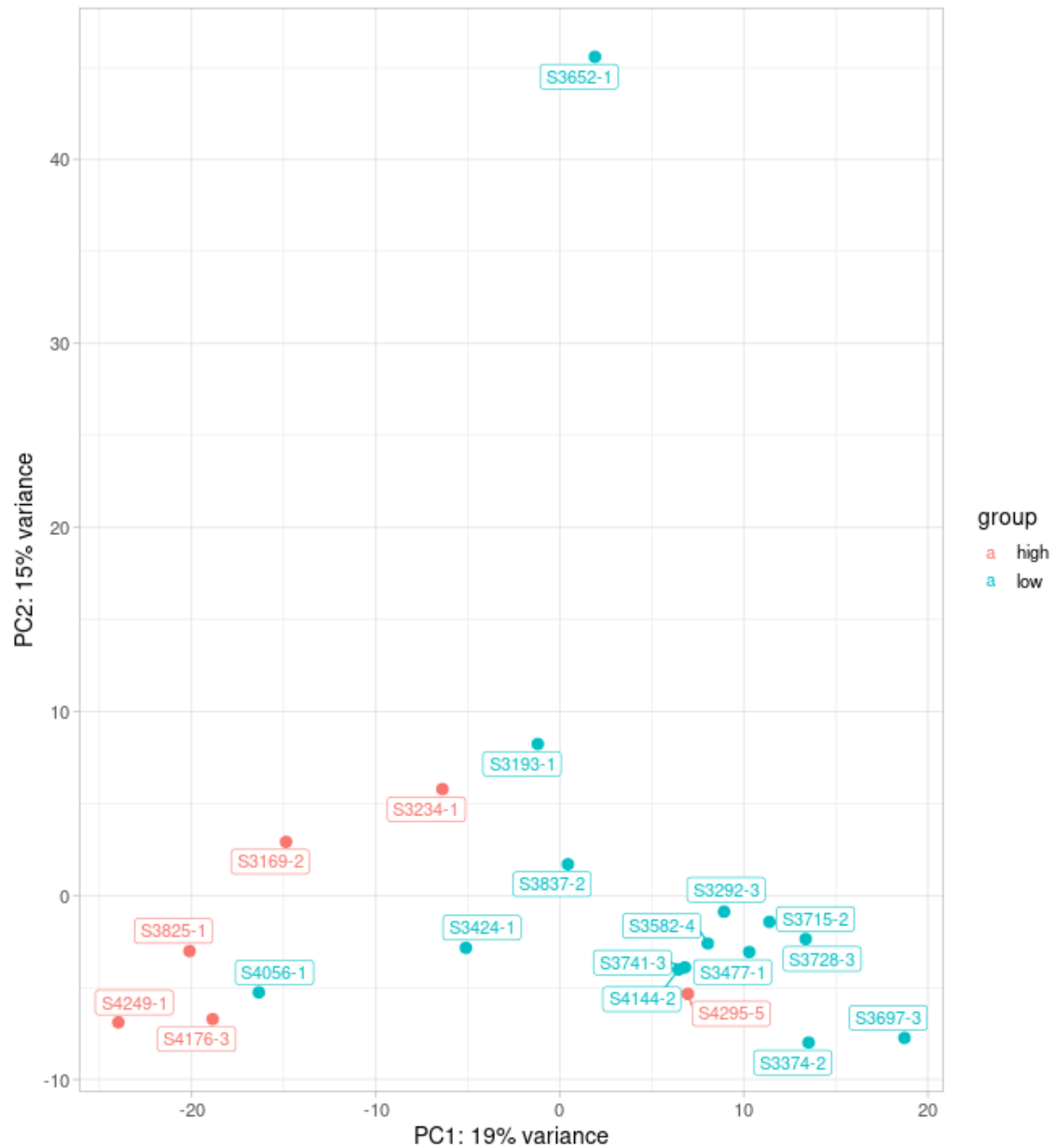
PCA - date_of

```r
# 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)
```

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)
```
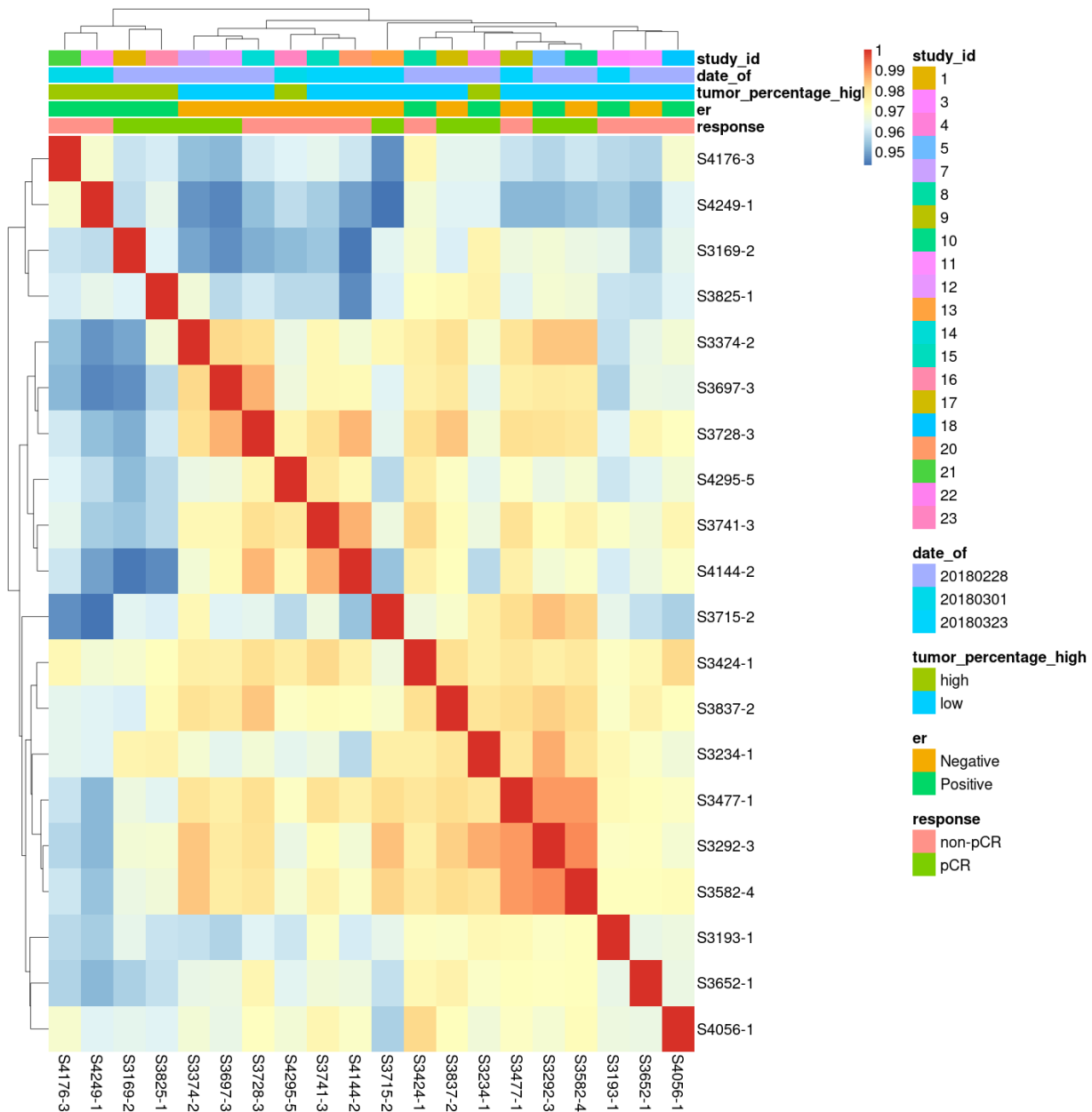
Response pCR vs non-pCR for Day 8 - see Table13

ER : Positive vs Negative for Day8 - Table 14

tumor_percentage_high : High vs Low for Day8- Table 15

date_of: 20180323 vs 20180228 - for Day8: Table 16

Visualization

Gene example

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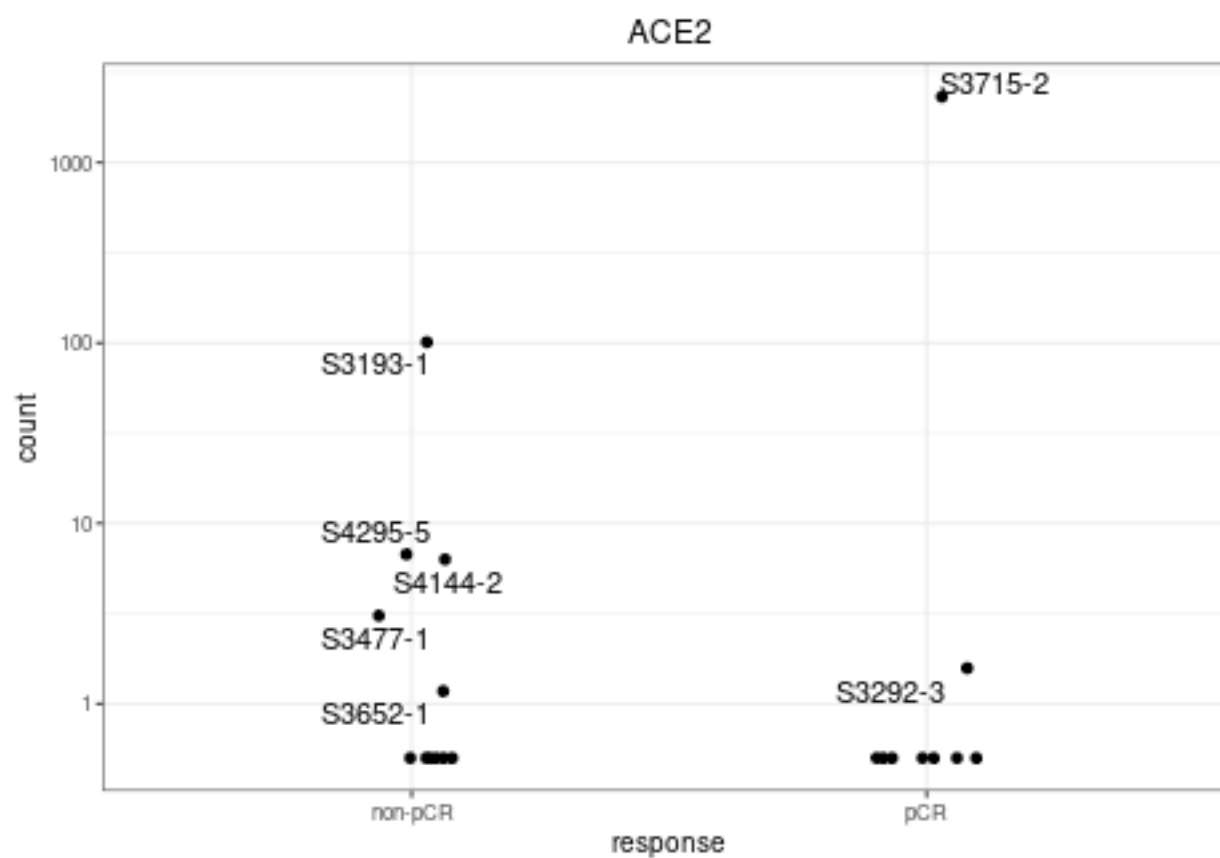
```r
# Add a column for significant genes
resResponse_tb_vis <- resResponse_tb %>% mutate(threshold = padj < 0.01)

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(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, 6))+
  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))
```

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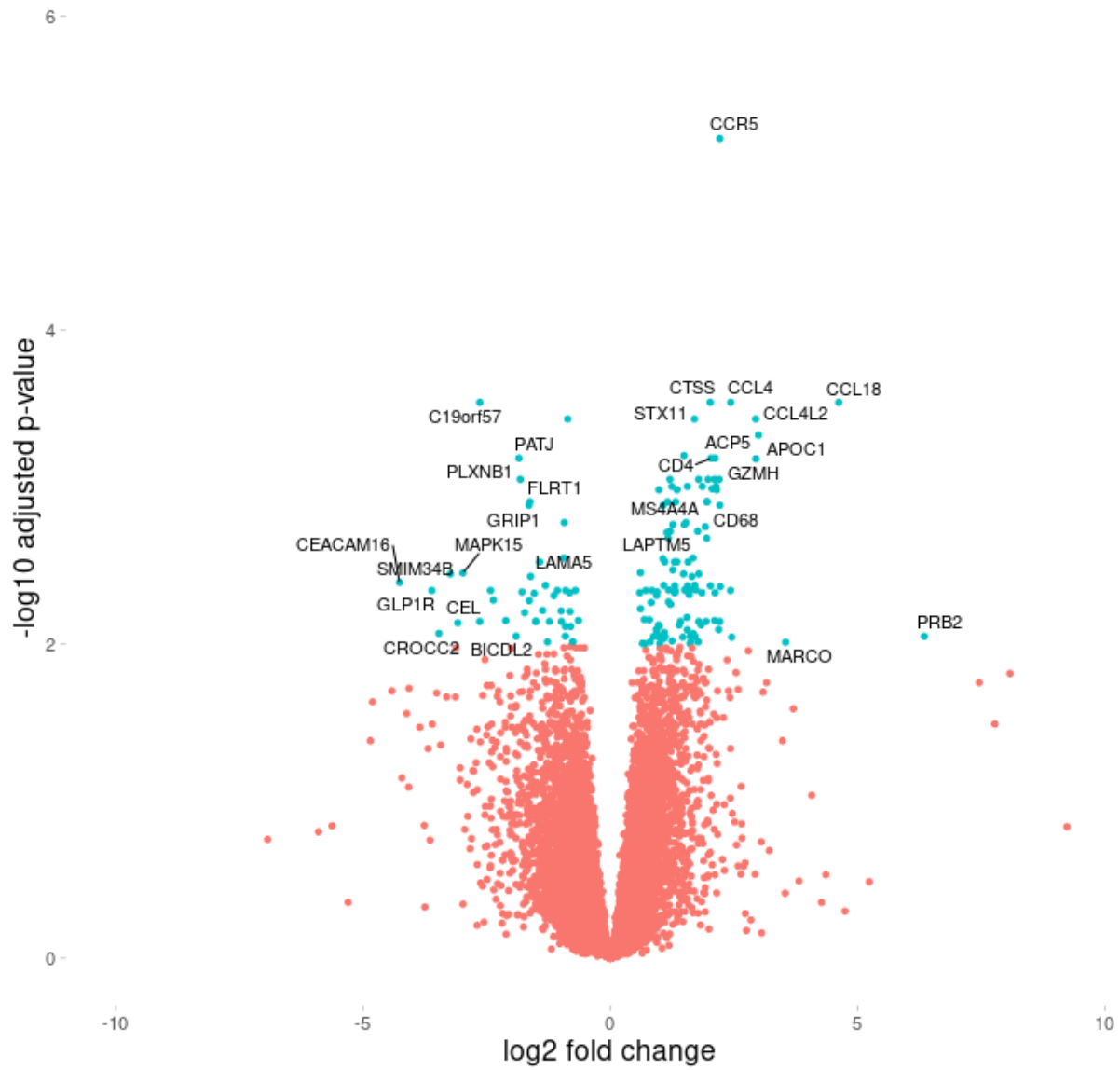
```
# Add a column for significant genes
resER_tb <- resER_tb %>% mutate(threshold = padj < 0.01)

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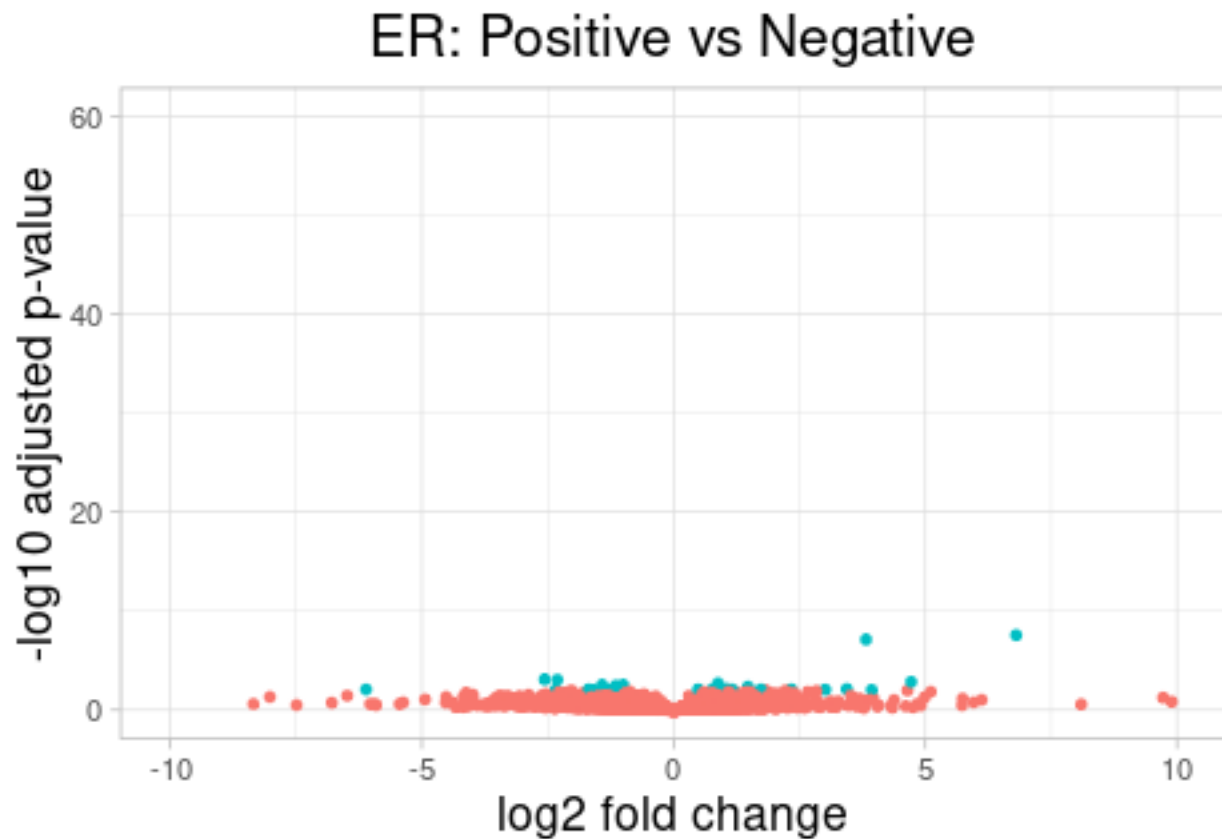
```
# Add a column for significant genes
resTP_tb <- resTP_tb %>% mutate(threshold = padj < 0.01)

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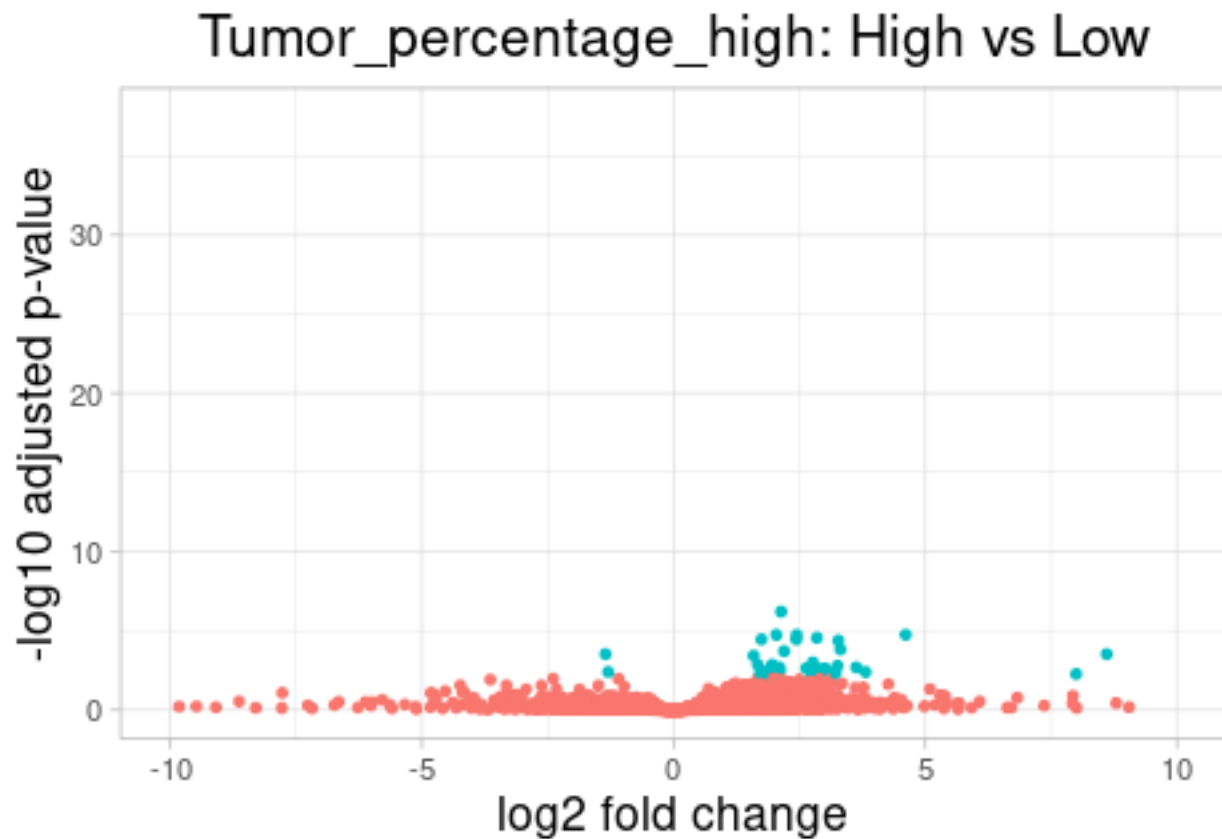
```r
# Add a column for significant genes
resDO_tb <- resDO_tb %>% mutate(threshold = padj < 0.01)

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)))
```
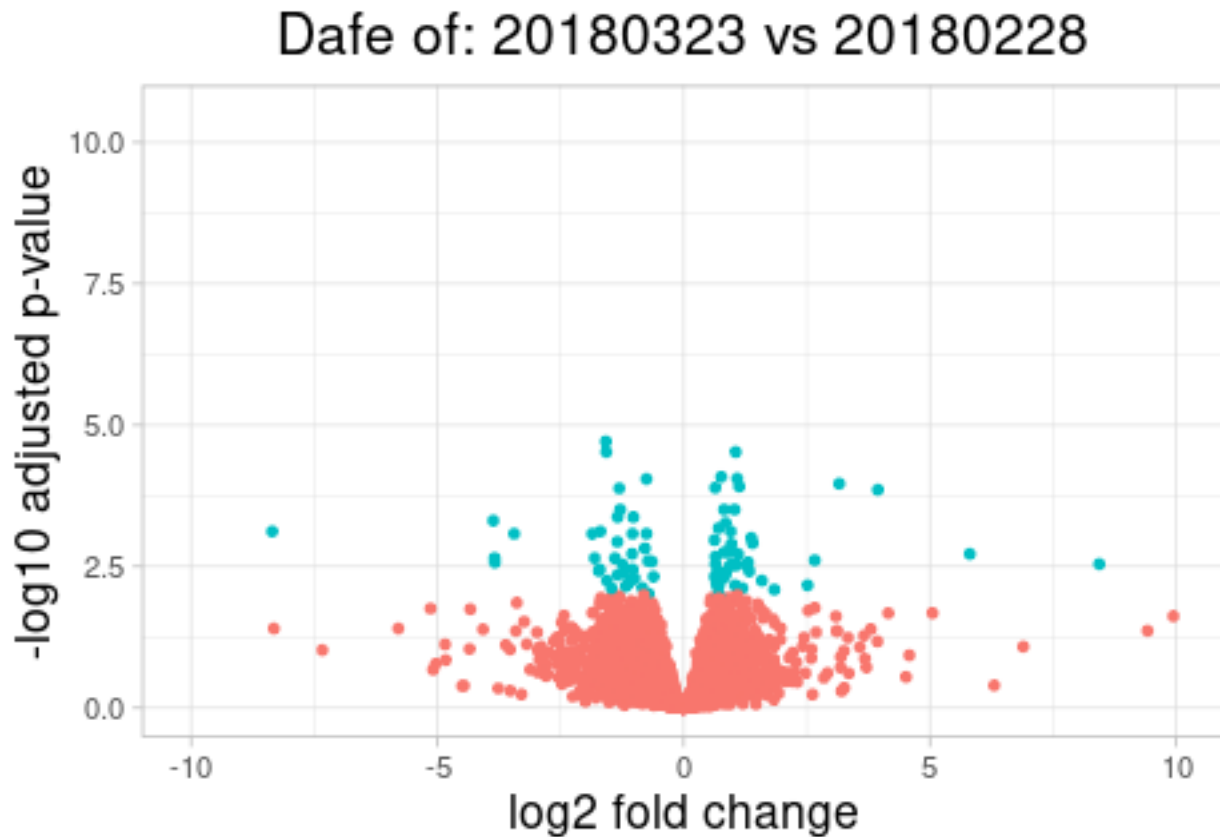
Heatmaps

```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

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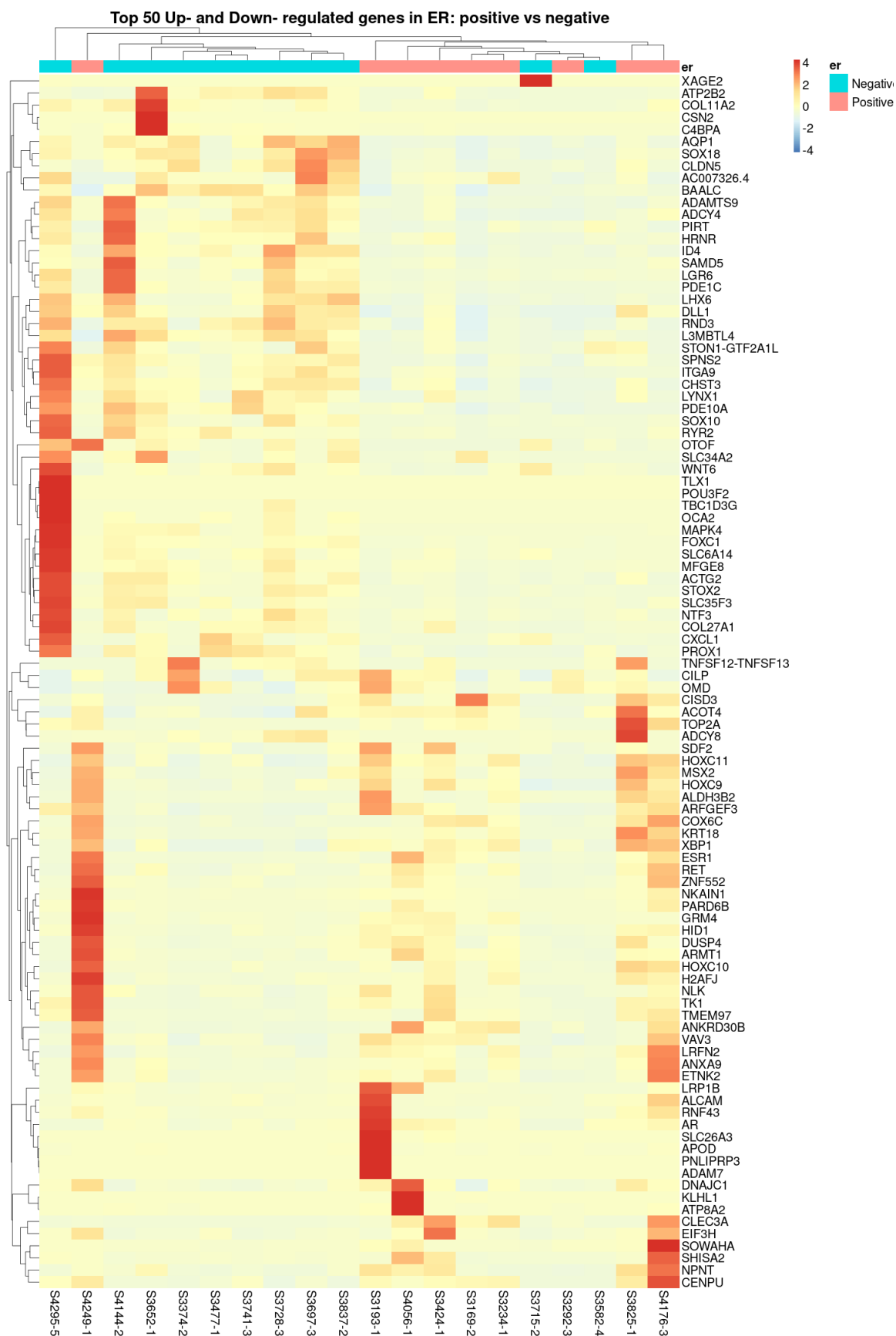
```r
# Create a matrix of normalized expression
sig_up <- resTP_tb_significant %>% arrange(-log2FoldChange) %>% head(50) %>% pull(gene)
sig_down <- resTP_tb_significant %>% arrange(log2FoldChange) %>% head(50) %>% pull(gene)
sig <- c(sig_up, sig_down)

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")

# 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)
```
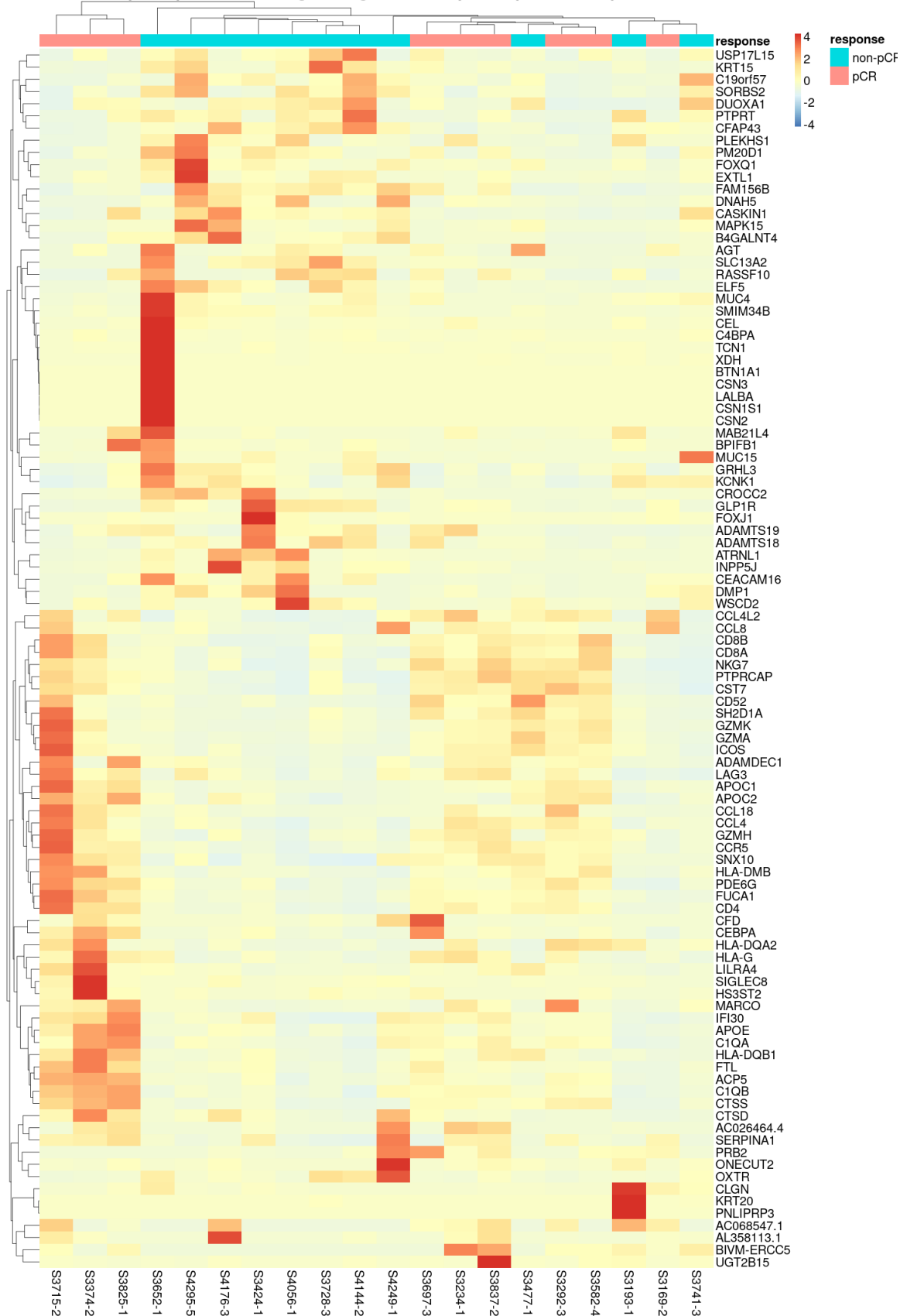
Top Up/Down-regulated genes in Tumor_percentage_high: high vs low

32

```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
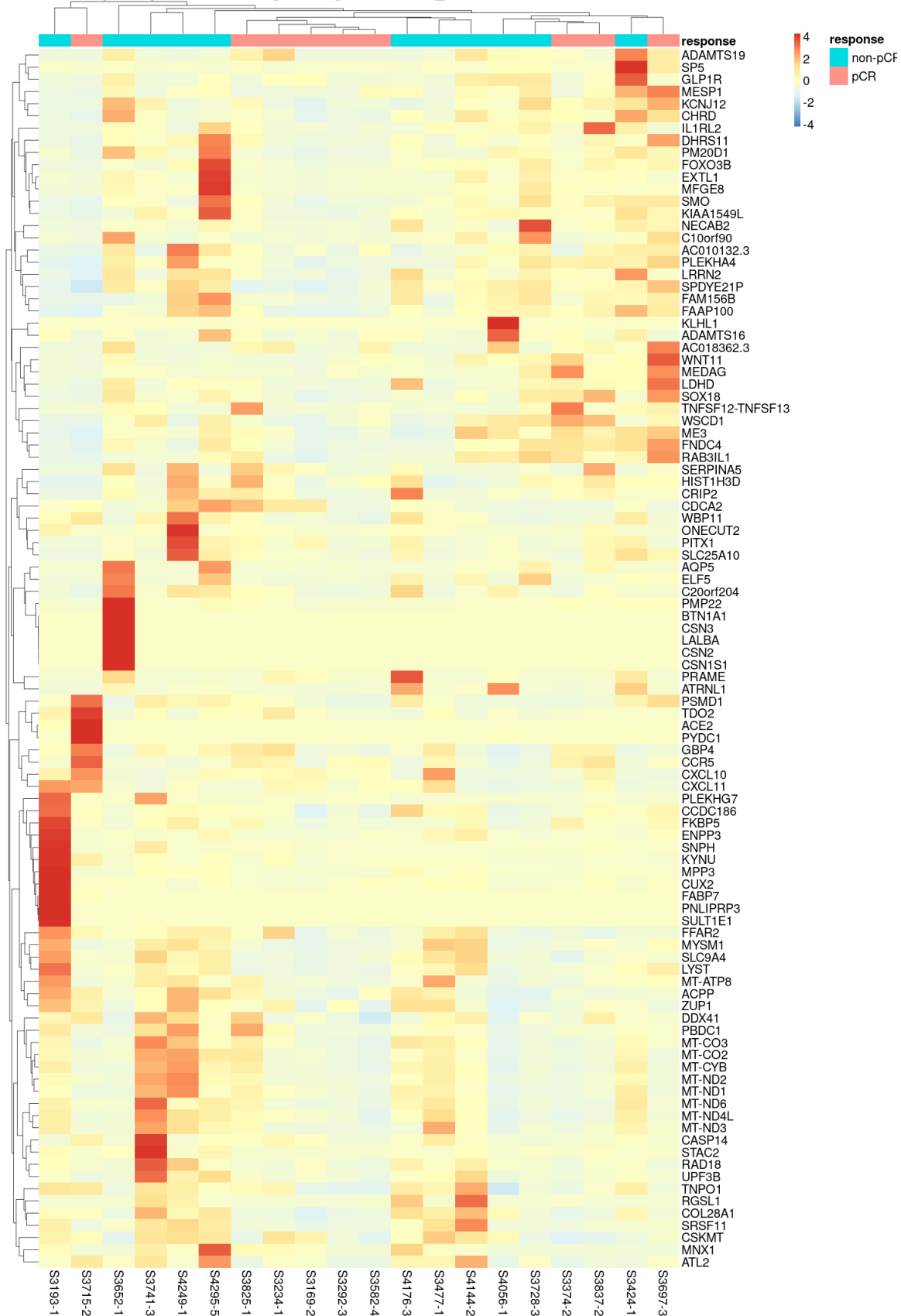
Functional analysis

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/T20.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 find

```
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 f
```

Molecular Function (MF)

```
bg_genes <- rownames(resResponse)

## Run GO enrichment analysis
compGO <- enrichGO(gene = sigResponse_up,
                   universe = bg_genes,
                   keyType = "ENSEMBL",
                   OrgDb = "org.Hs.eg.db",
                   ont = "MF",
                   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_MF_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 find
```
```
results_down <- data.frame(compGO@result) %>% dplyr::filter(p.adjust < 0.05)
```

```
## Error in data.frame(compGO@result): object 'compGO' not found
```
```r
print("Down")
```
```
## [1] "Down"
```
```r
nrow(results_down)
```
```
## Error in h(simpleError(msg, call)): error in evaluating the argument 'x' in selecting a method for f
```

Cellular Compartment (CC)

```r
bg_genes <- rownames(resResponse)

## Run GO enrichment analysis
compGO <- enrichGO(gene = sigResponse_up,
                   universe = bg_genes,
                   keyType = "ENSEMBL",
                   OrgDb = "org.Hs.eg.db",
                   ont = "CC",
                   qvalueCutoff  = 0.05,
                   pAdjustMethod = "BH",
                   readable = TRUE)
```
```
## Error in enrichGO(gene = sigResponse_up, universe = bg_genes, keyType = "ENSEMBL", : could not find
```
```r
#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"
```
```r
results_up <- data.frame(compGO@result) %>% dplyr::filter(p.adjust < 0.05)
```
```
## Error in data.frame(compGO@result): object 'compGO' not found
```
```r
nrow(results_up)
```
```
## Error in h(simpleError(msg, call)): error in evaluating the argument 'x' in selecting a method for fu
```
```r
write_csv(results_up, "tables/T22.day8.GO_CC_UP.csv")
```
```
## Error in is.data.frame(x): object 'results_up' not found
```
```r
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 find
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] org.Hs.eg.db_3.12.0       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                    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
```