DE analysis - Day8

Sergey Naumenko

2021-05-09

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

```
# 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"
```

```
attributes(txi)
```

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

```
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    22
```

Sanity check that metadata matches your expression

It is always a good idea to check if:

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

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

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

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

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

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

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

Run DESeq2

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

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

Wald test

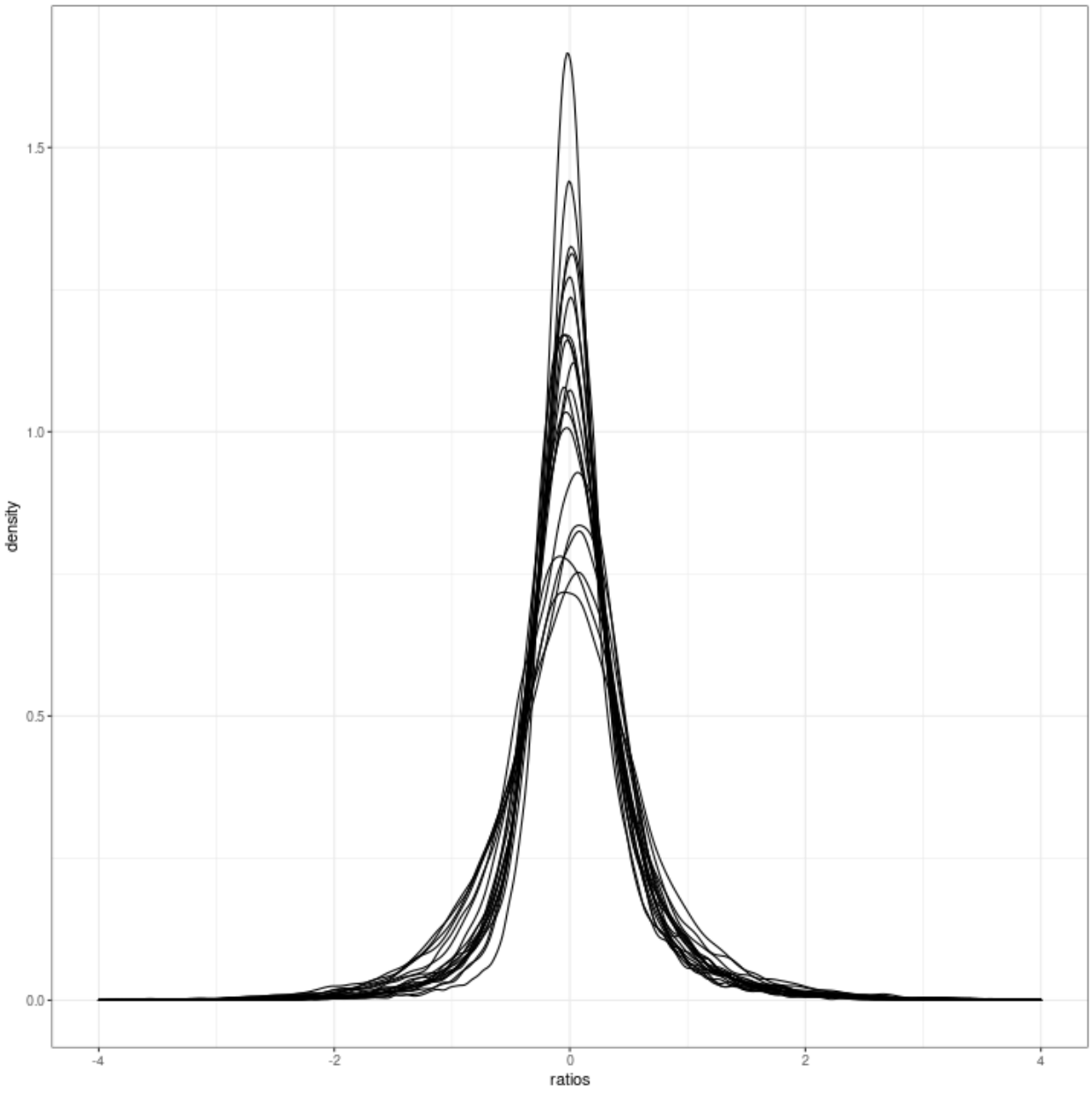
Here we subset protein coding genes.

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

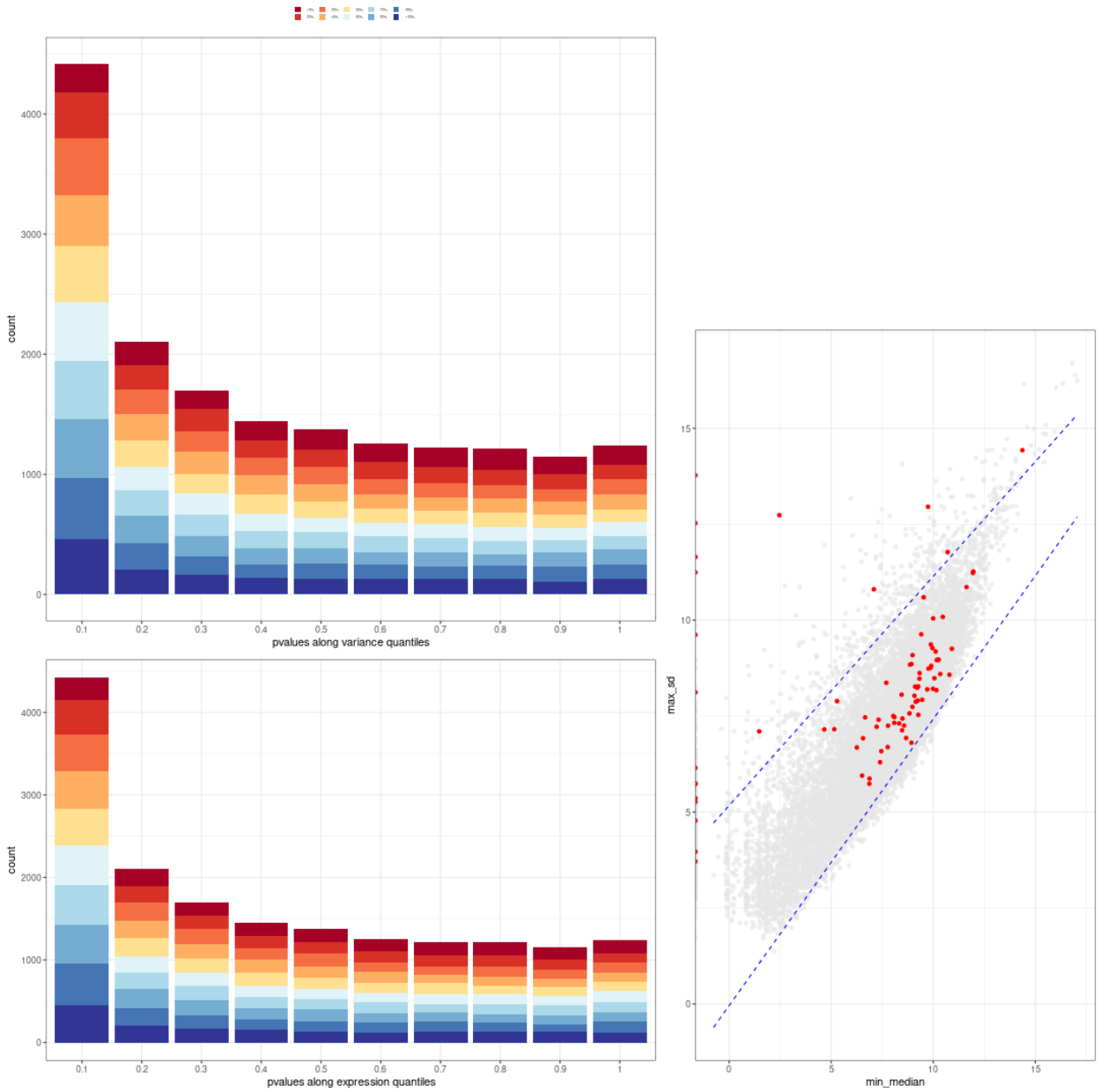
```
counts <- counts(dds, normalized = TRUE)
design <- as.data.frame(colData(dds))
degCheckFactors(counts[, 1:20])
```



Mean-Variance QC plots

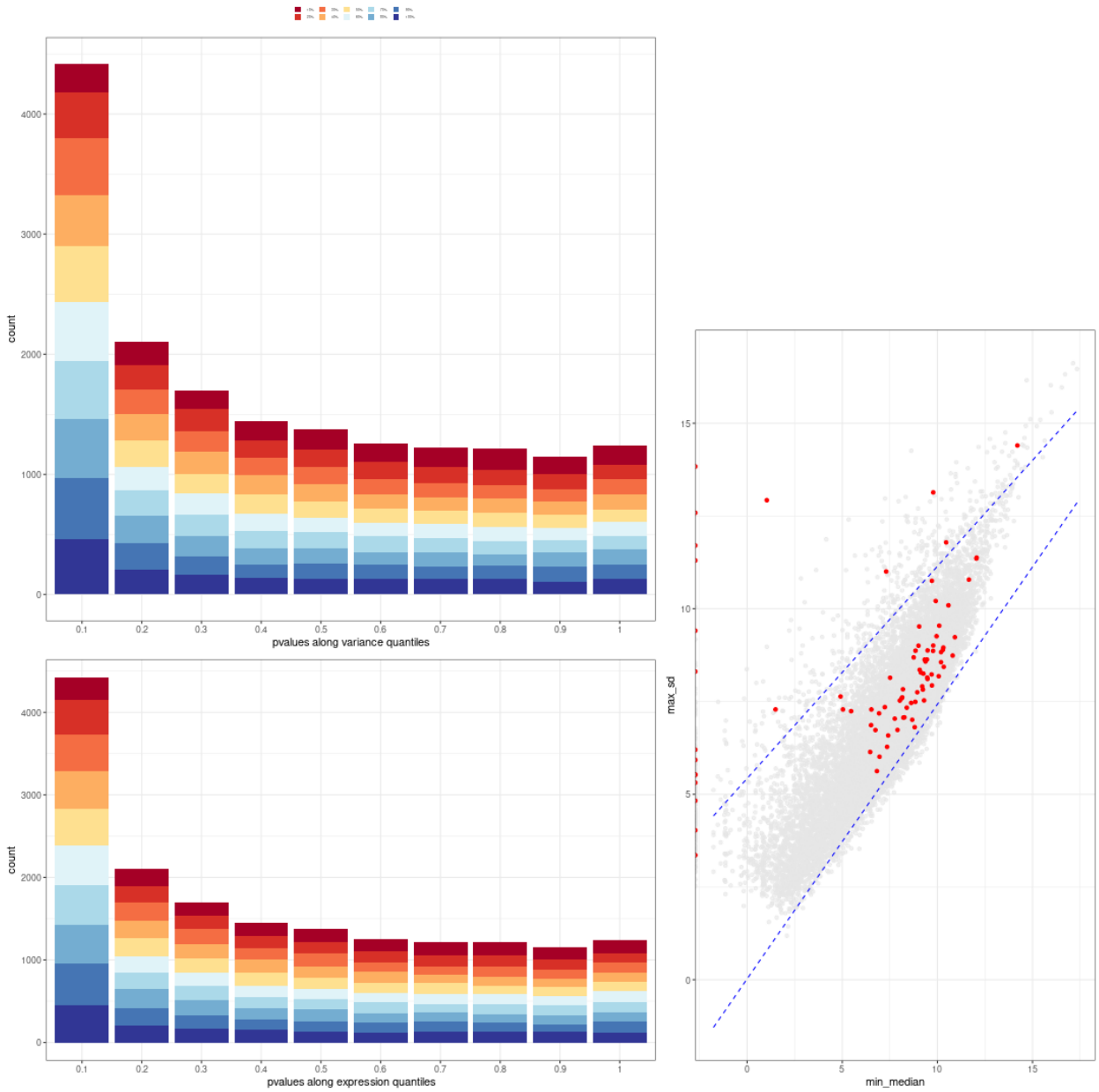
response

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



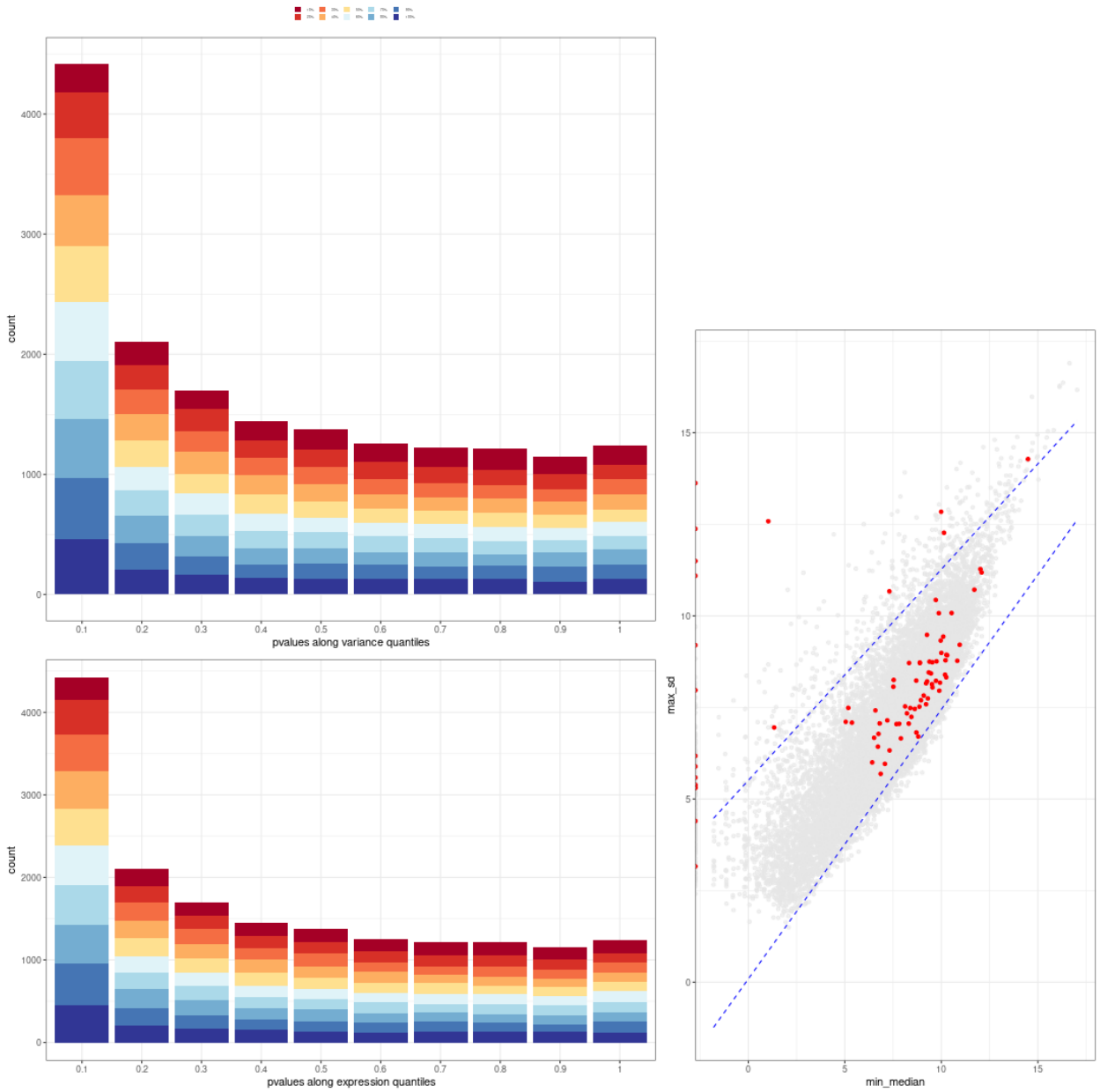
ER

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



tumor_percentage_high

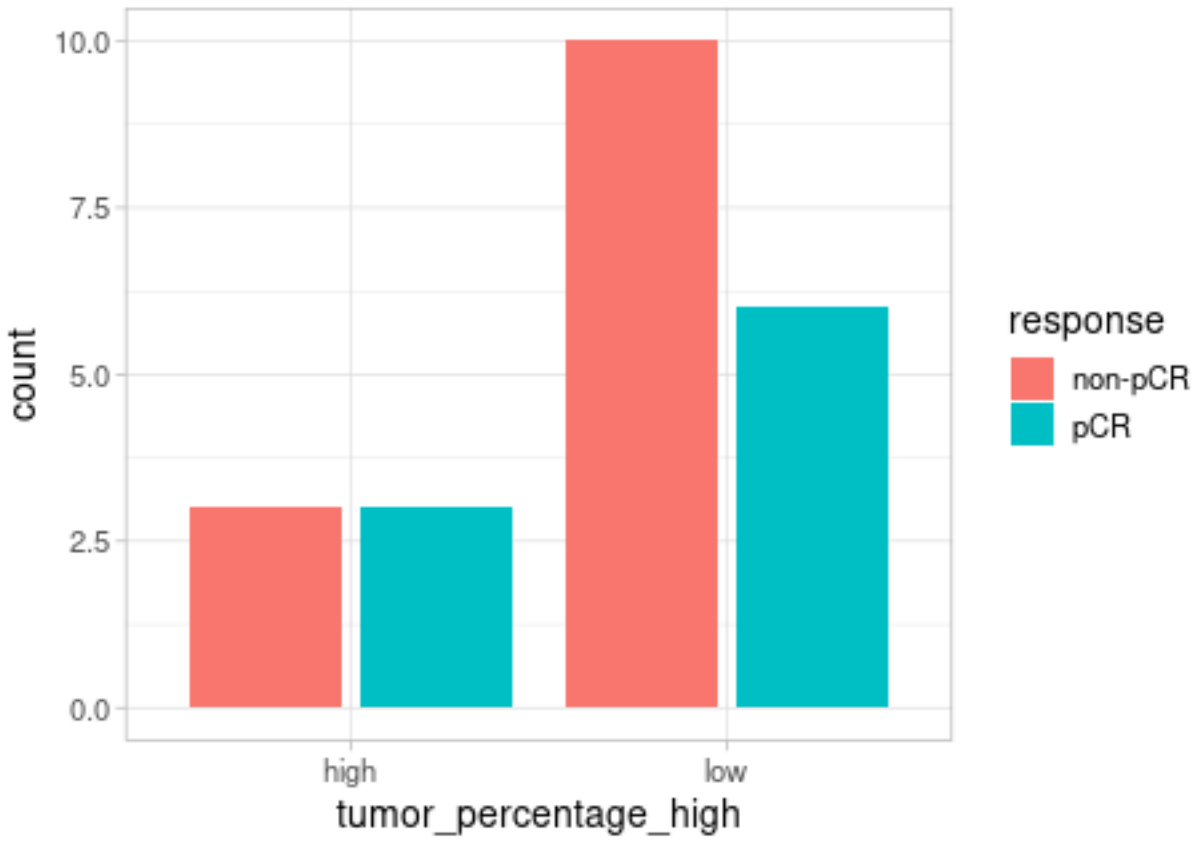
```
degQC(counts, design[["tumor_percentage_high"]], pvalue = res[["pvalue"]])
```



Covariates effect on count data

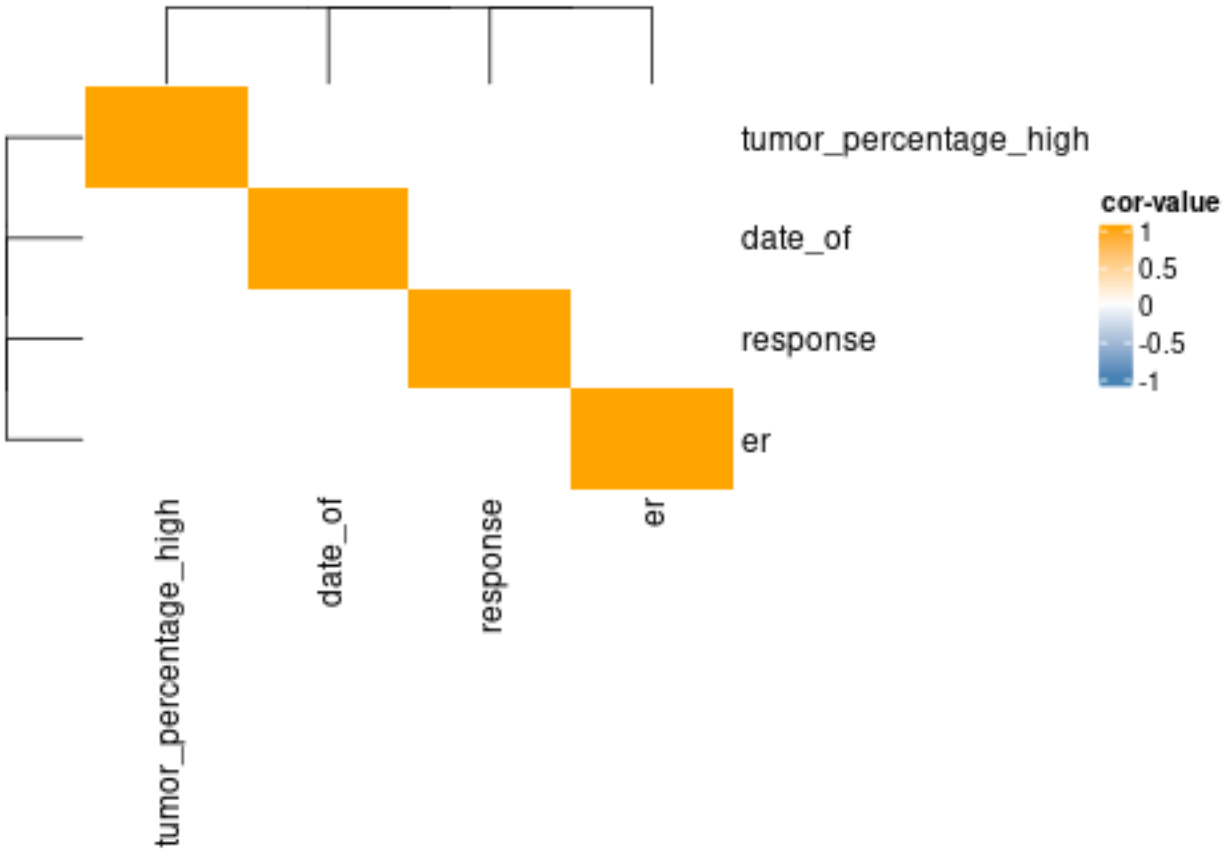
```
mdata <- colData(dds) %>%  as.data.frame()  %>%
  dplyr::select(response, er, date_of, tumor_percentage_high)

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

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

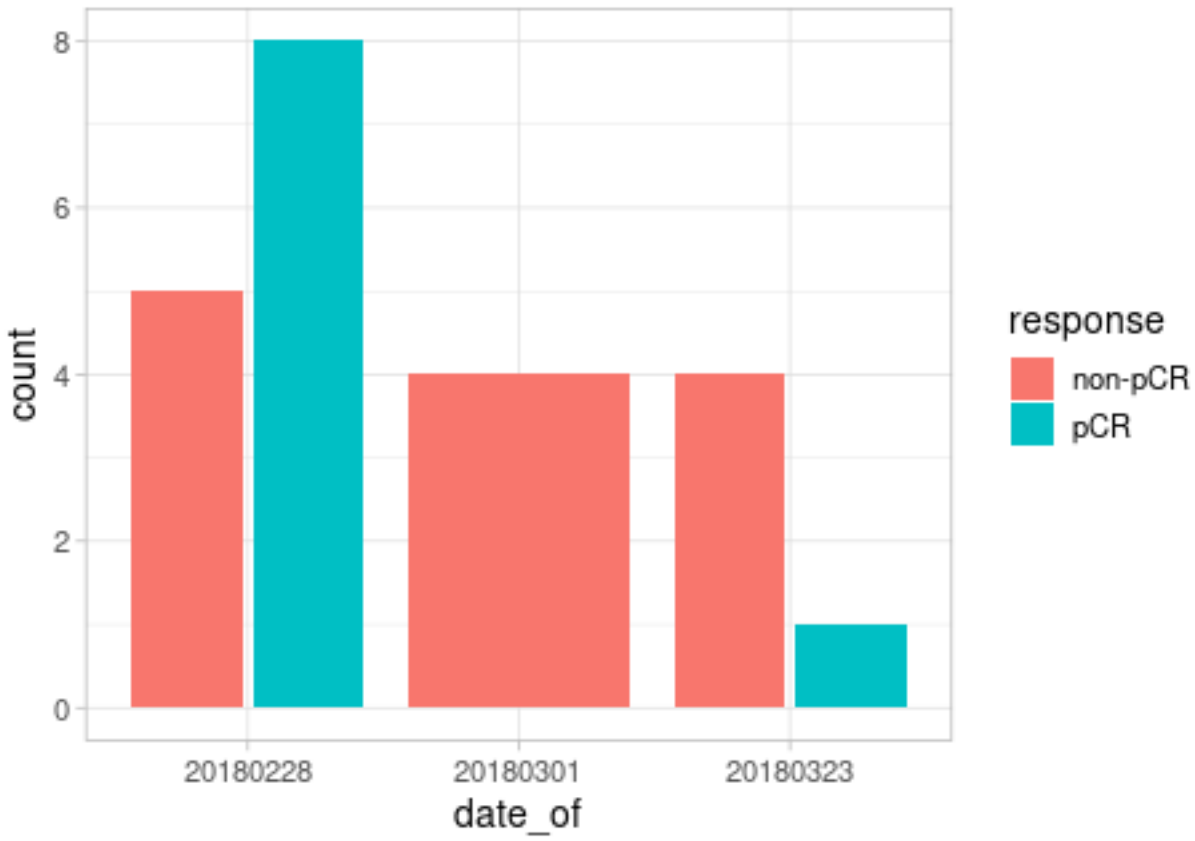


Covariates correlation with metrics

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




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



Sample-level QC analysis

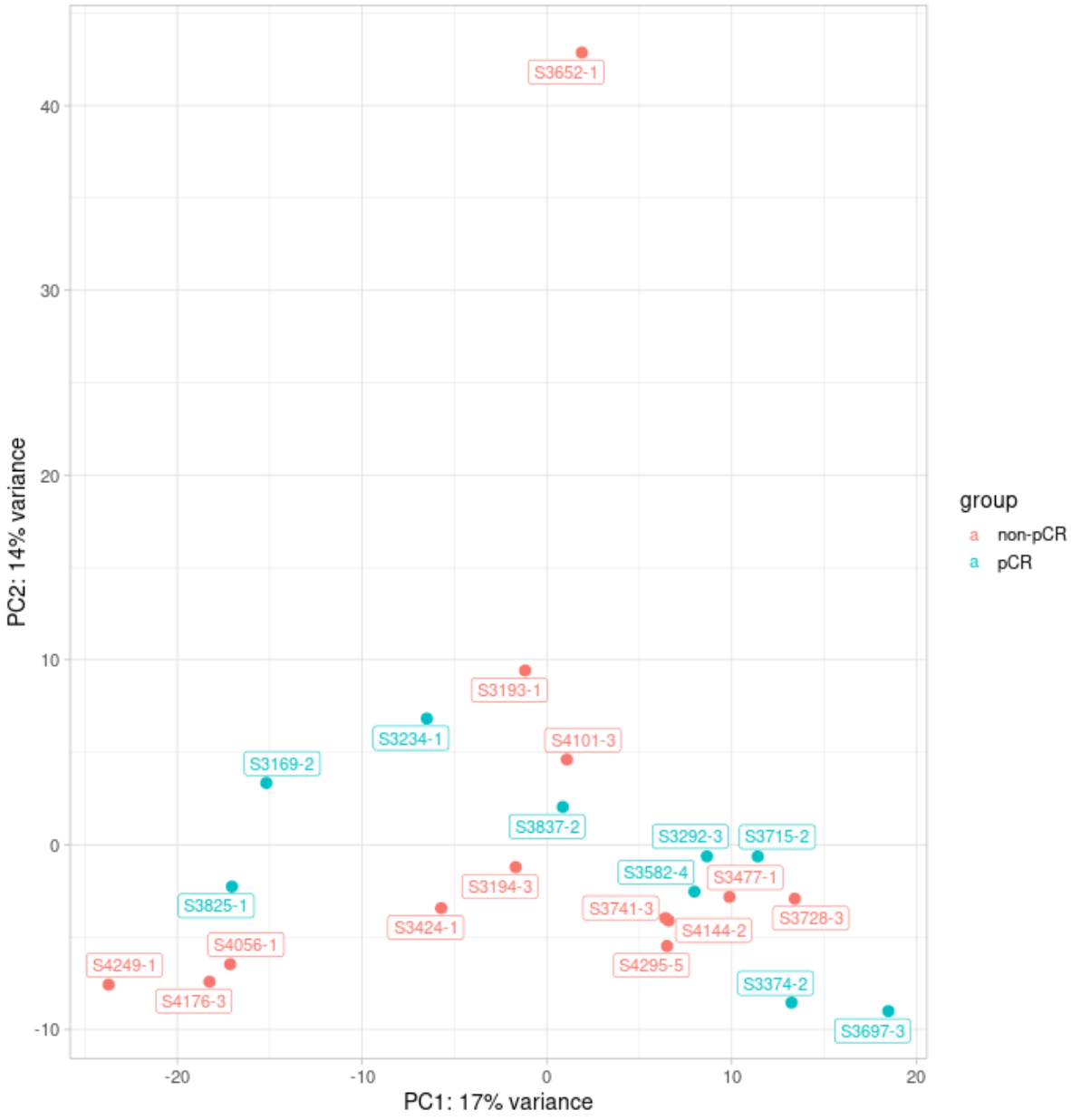
```
### Transform counts for data visualization (unsupervised analysis)
rld_file <- "data/rld.day8.RDS"
if (!rebuild_rds & file.exists(rld_file)){
    rld <- readRDS(rld_file)
}else{
    rld <- rlog(dds, blind = TRUE)
    saveRDS(rld, rld_file)
}
class(rld) # what type of object is this
```

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

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

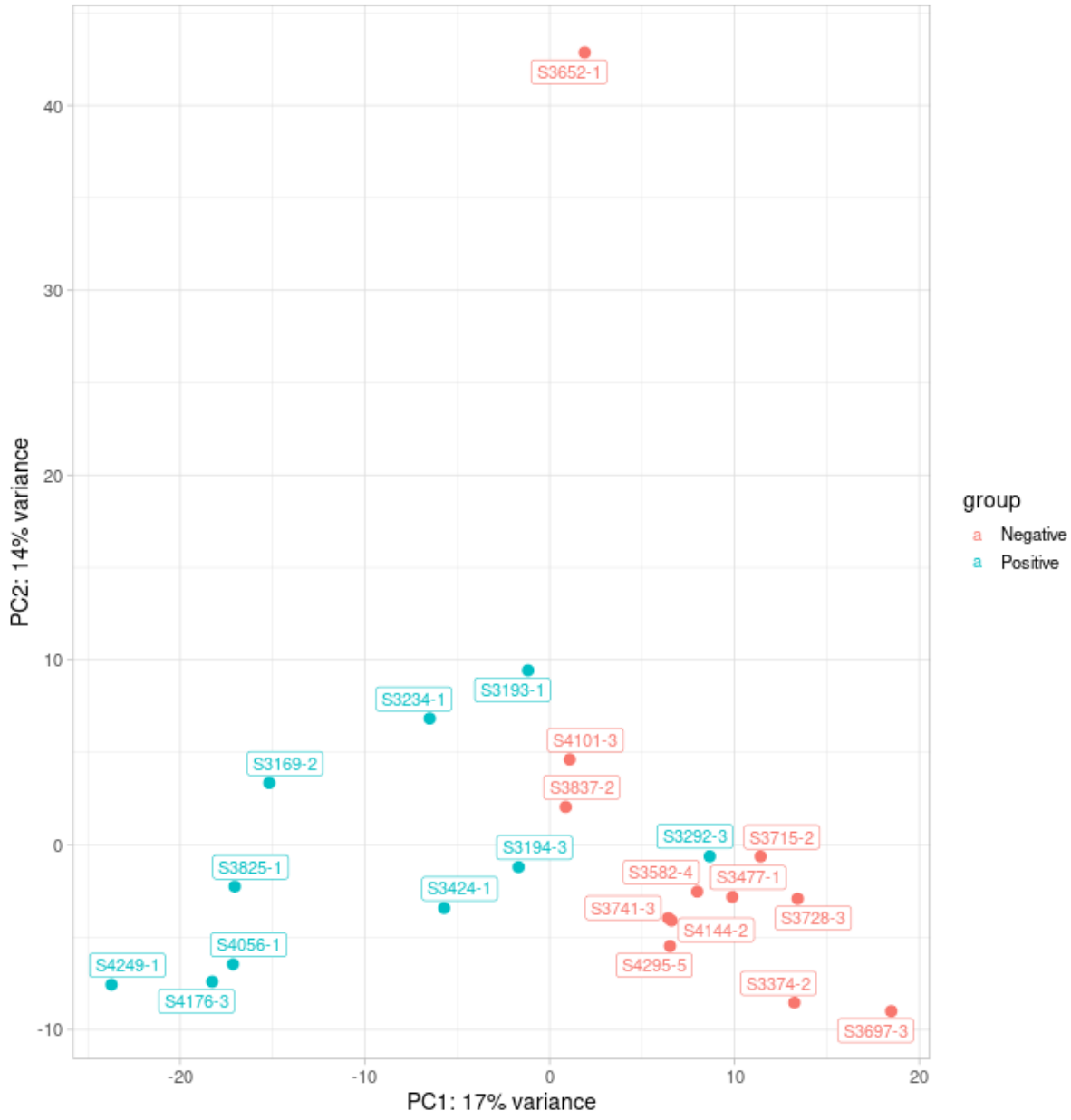
PCA - response

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



PCA - ER

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

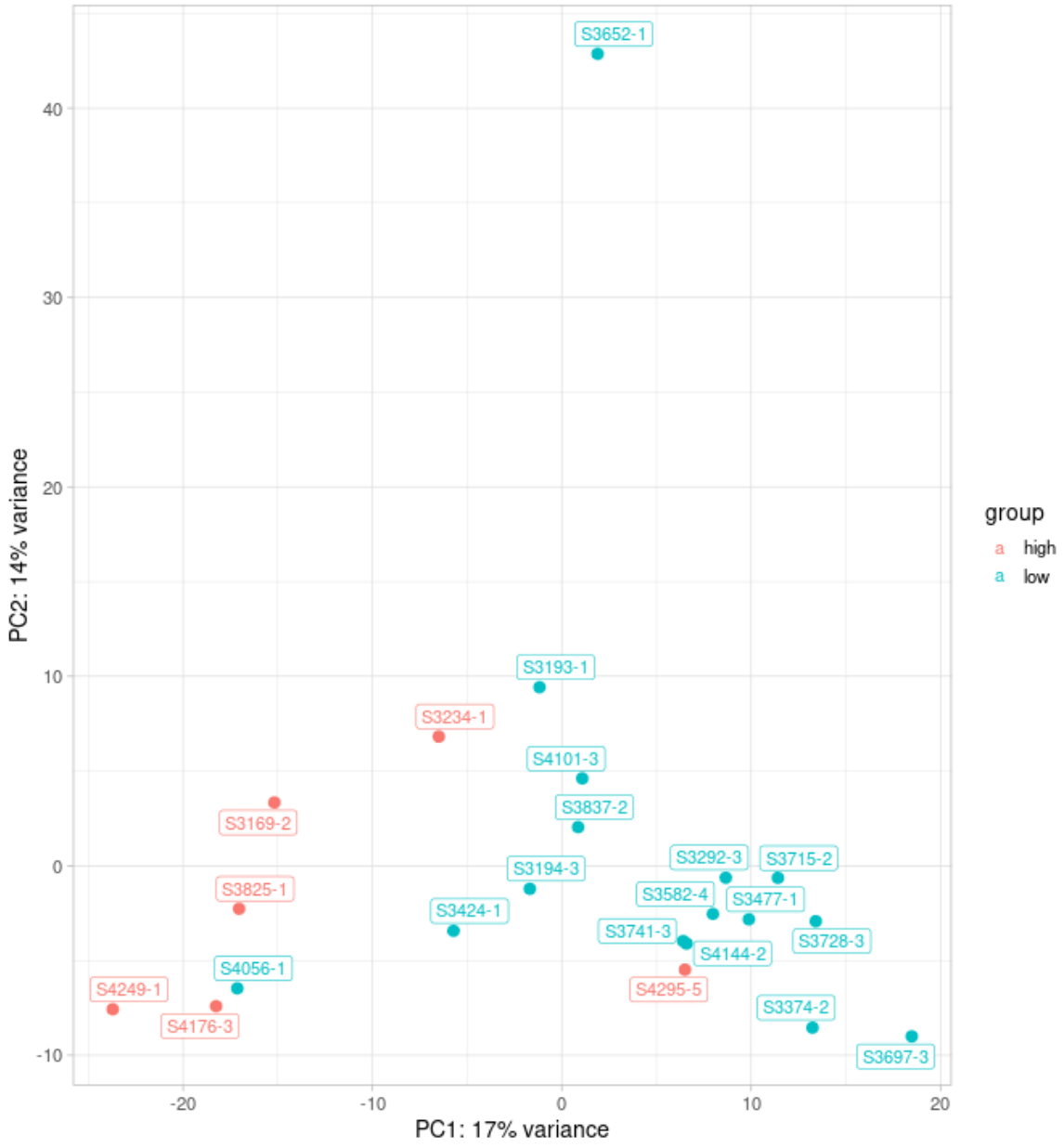


PCA - tumor_percentage

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

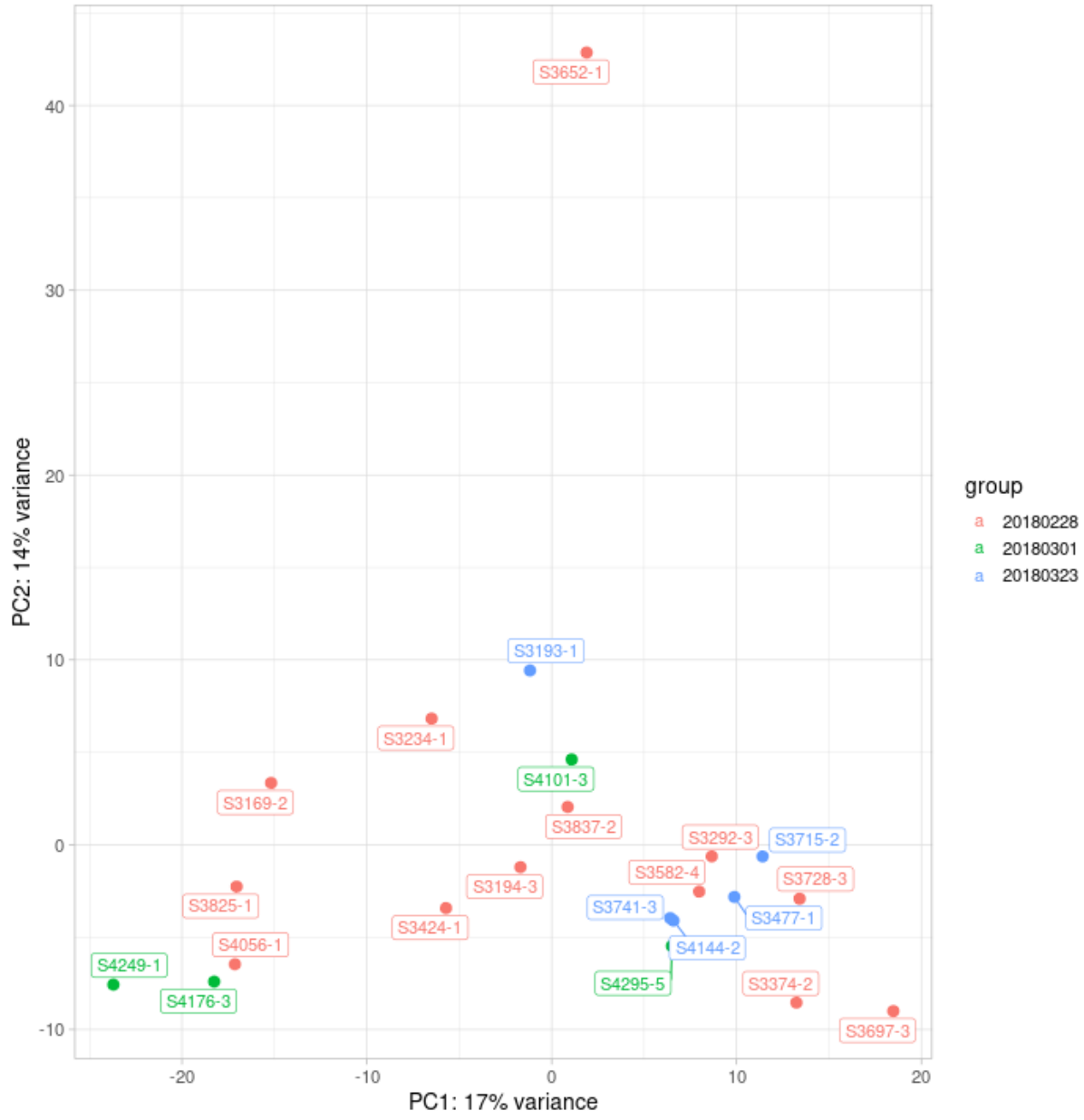
PCA - tumor_percentage_high

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



PCA - date_of

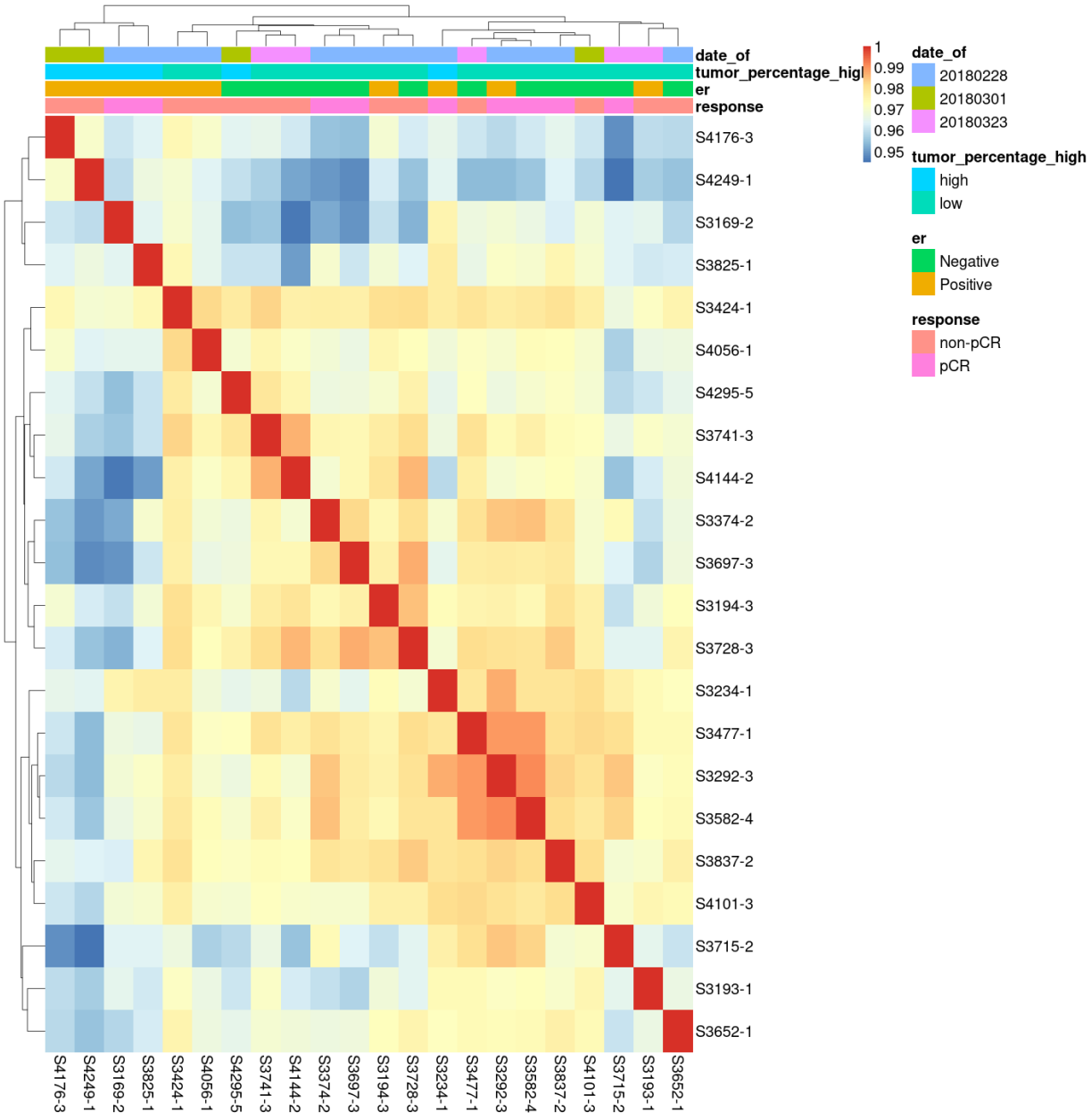
```
# Use the DESeq2 function
plotPCA(rld, intgroup = c("date_of")) + geom_label_repel(aes(label = name))
```



Inter-correlation analysis

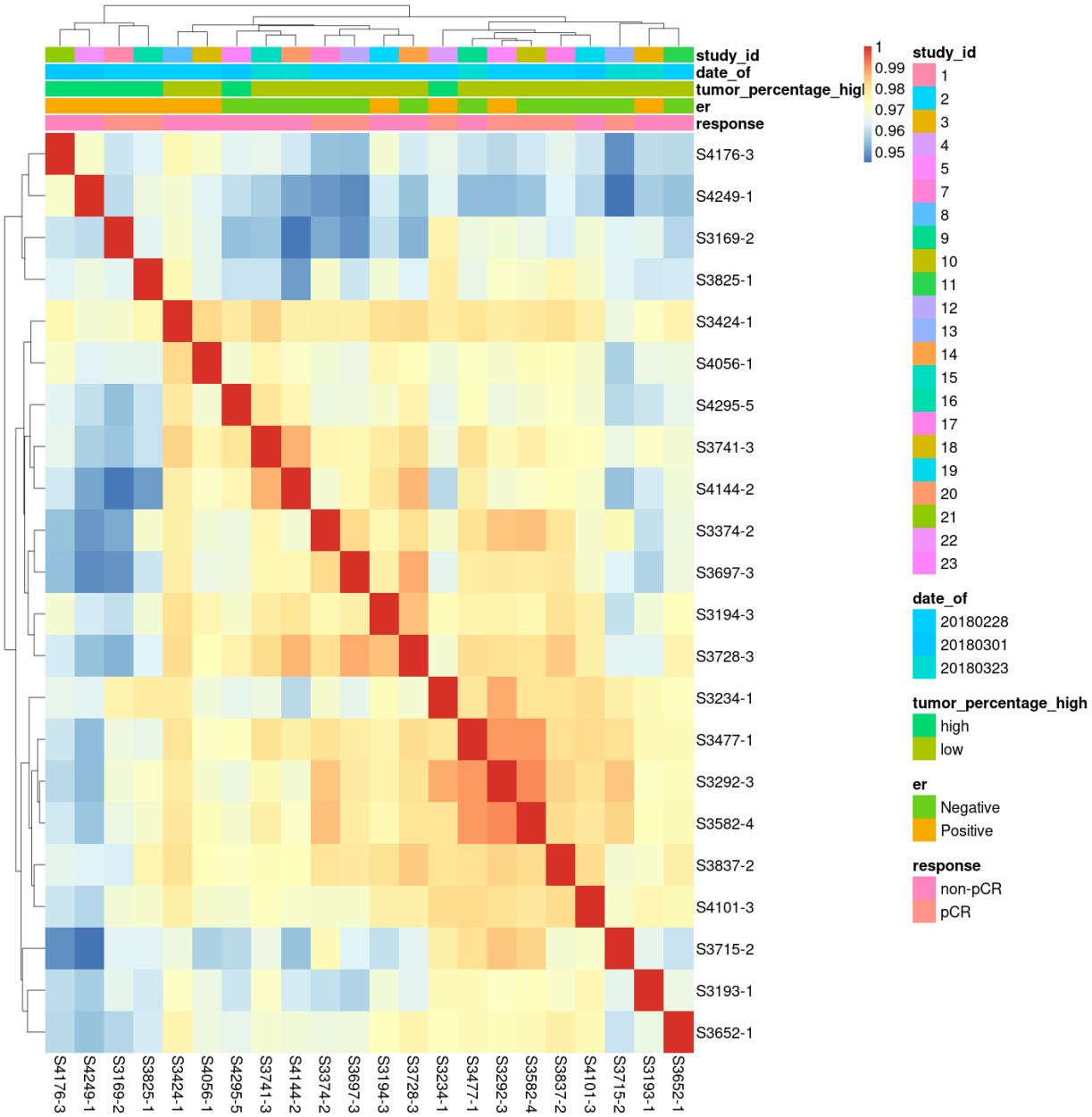
Without study_id

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

meta$study_id <- as.factor(meta$study_id)
# Create annotation file for samples
annotation <- meta[, c("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
S4176-3
S4249-1
S3169-2
S3825-1
S3424-1
S4056-1
S4295-5
S3741-3
S4144-2
S3374-2
S3697-3
S3194-3
S3728-3
S3234-1
S3477-1
S3292-3
S3582-4
S3837-2
S4101-3
S3715-2
S3193-1
S3652-1
1
0.99
0.98
0.97
0.96
0.95
date_of
20180228
20180301
20180323
tumor_percentage_high
high
low
er
Negative
Positive
response
non-pCR
pCR

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("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

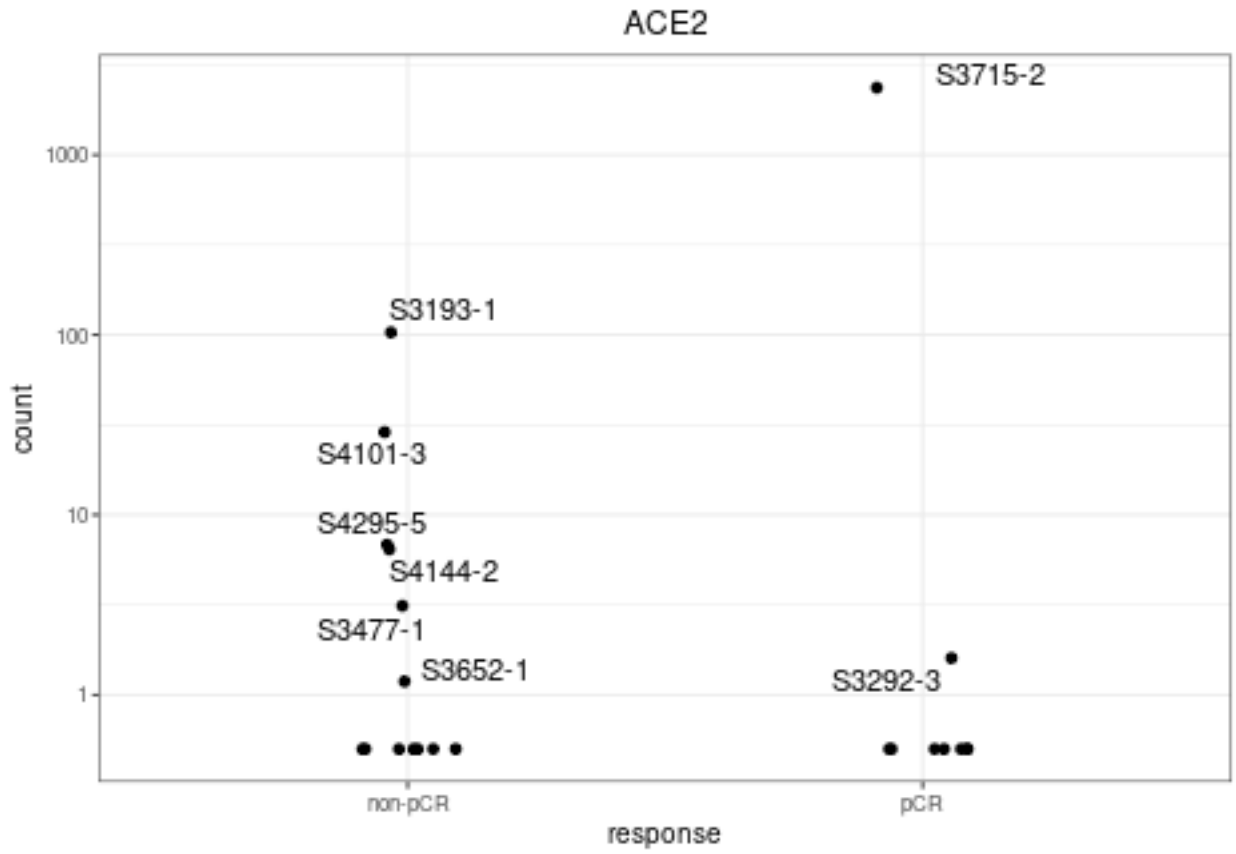
tumor_percentage_high : High vs Low for Day8- Table 15

date_of: 20180323 vs 20180228 - for Day8: Table 16

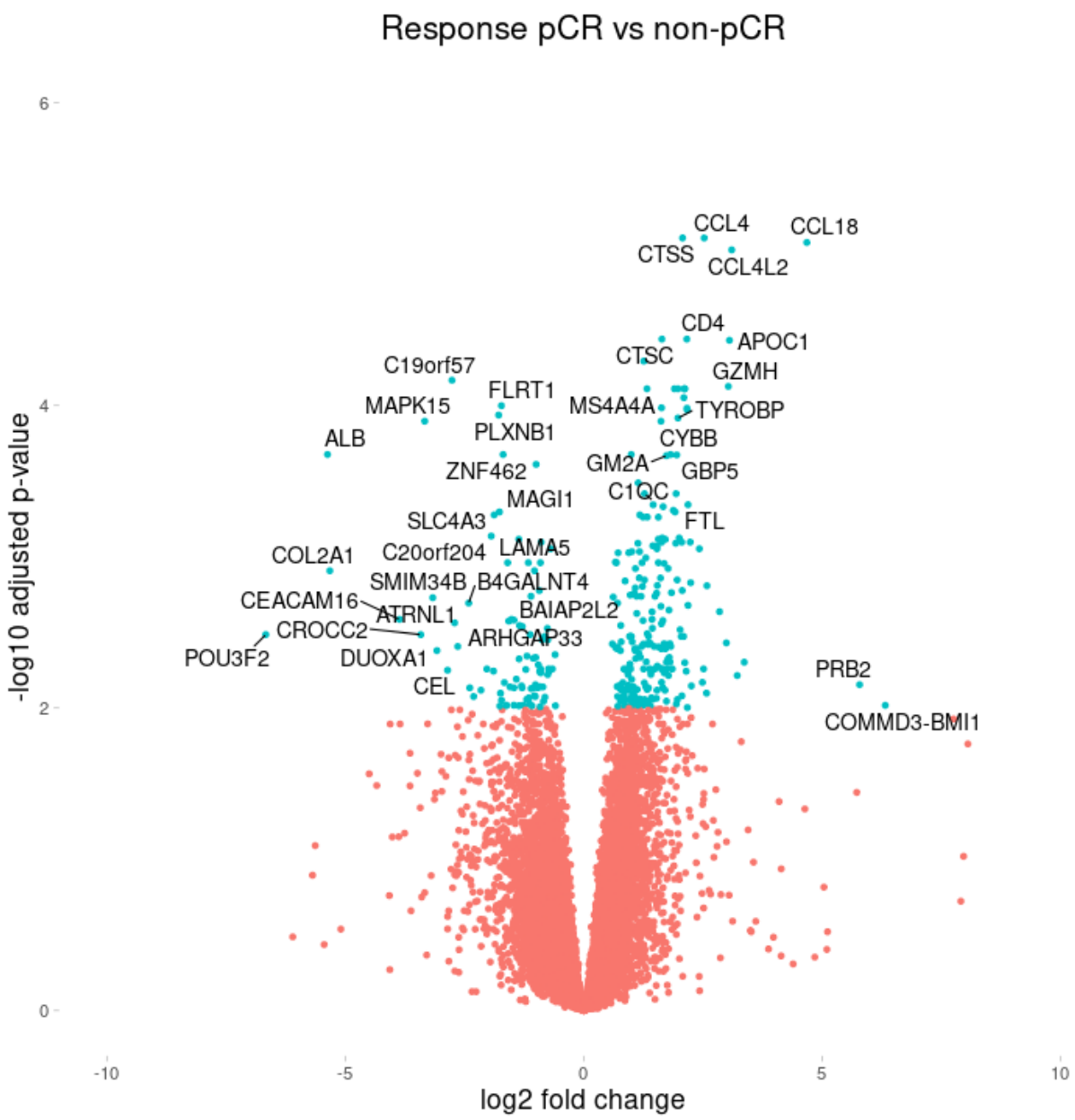
Visualization

Gene example

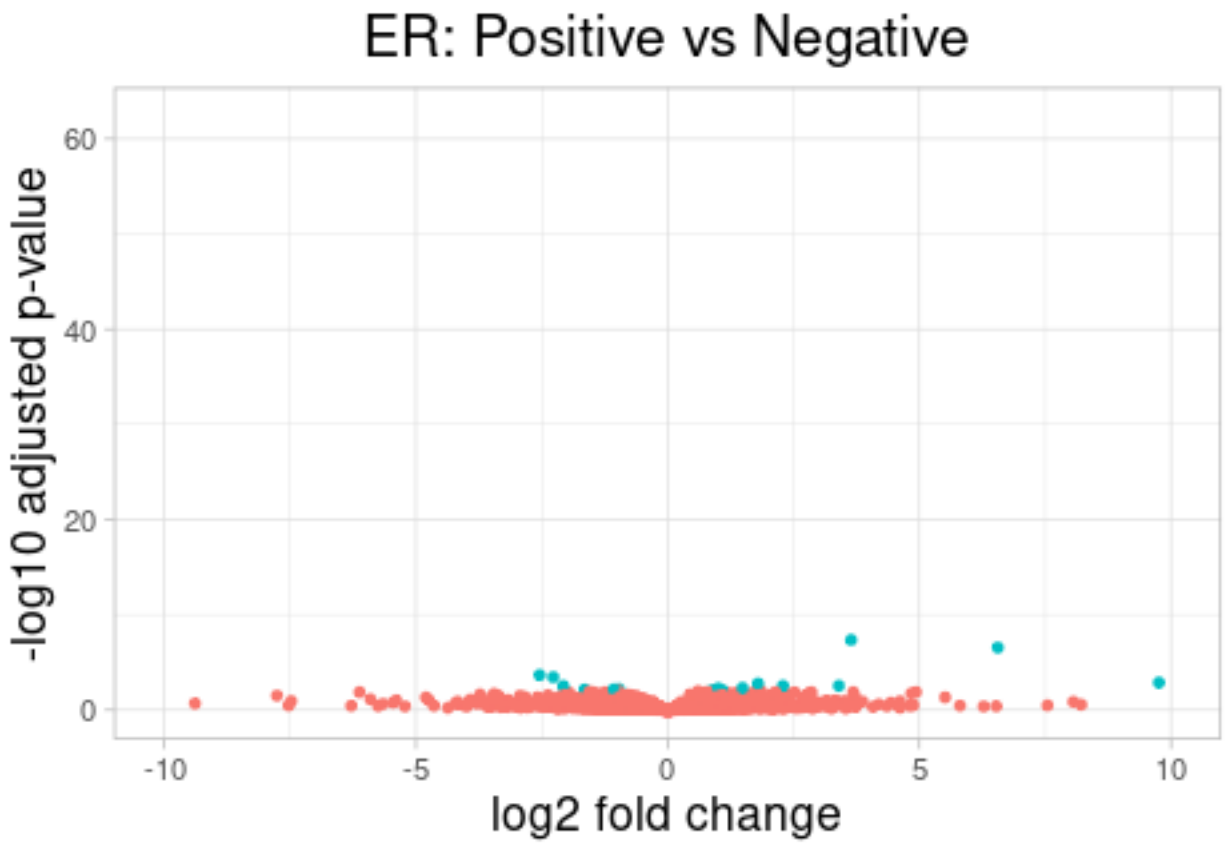
```
d <- plotCounts(dds,
                gene = "ENSG00000130234",
                intgroup = "response",
                returnData = TRUE)

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()
```



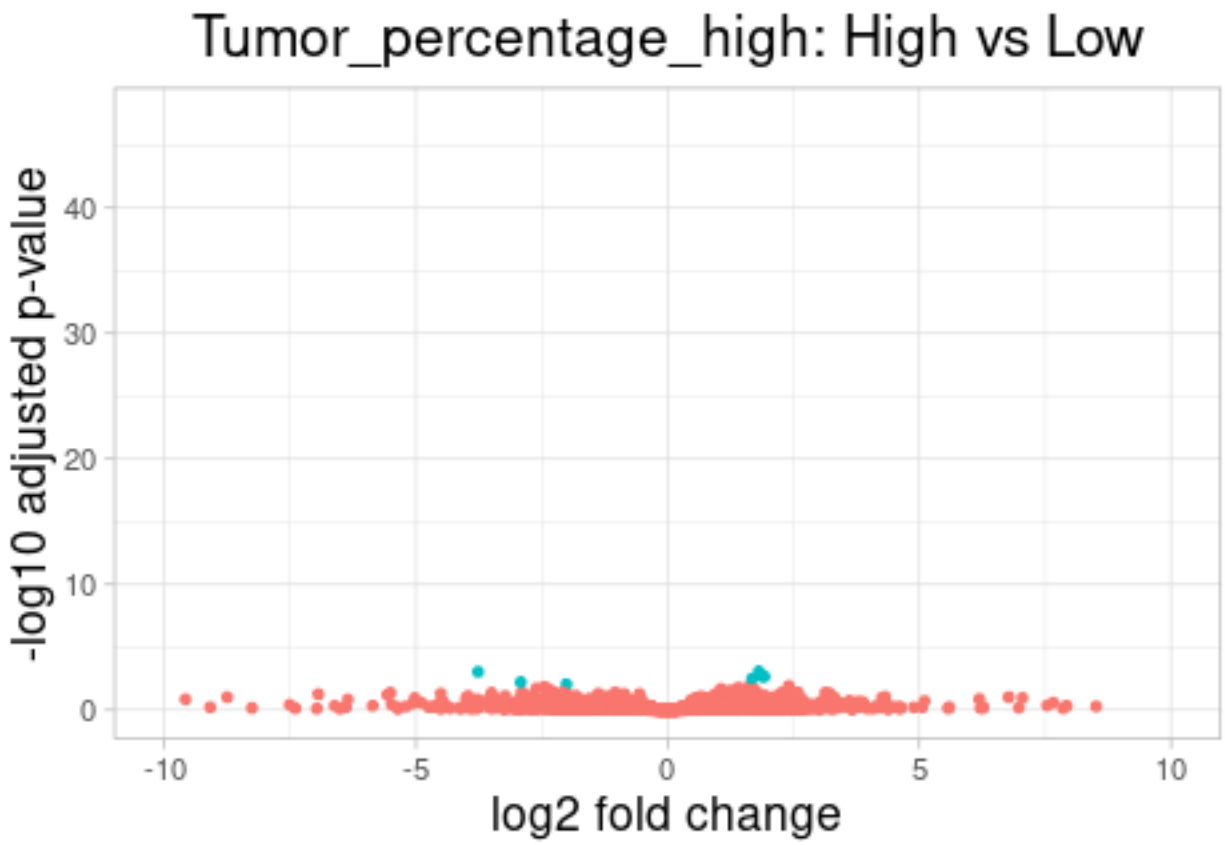
```r
# Add a column for significant genes
resResponse_tb_vis <- resResponse_tb %>% mutate(threshold = padj < 0.01)

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)

fig4b <- 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), size = 5)

saveRDS(fig4b, "data/fig4b.RDS")
fig4b
```


Response pCR vs non-pCR
-log10 adjusted p-value
log2 fold change
6
4
2
0
-10
-5
0
5
10
CCL4
CCL18
CTSS
CCL4L2
CD4
APOC1
CTSC
GZMH
C19orf57
FLRT1
MAPK15
MS4A4A
TYROBP
PLXNB1
CYBB
ALB
GM2A
GBP5
ZNF462
C1QC
MAGI1
FTL
SLC4A3
COL2A1
C20orf204
LAMA5
SMIM34B
B4GALNT4
CEACAM16
ATRNL1
BAIAP2L2
CROCC2
ARHGAP33
POU3F2
DUOXA1
CEL
PRB2
COMMD3-BMI1

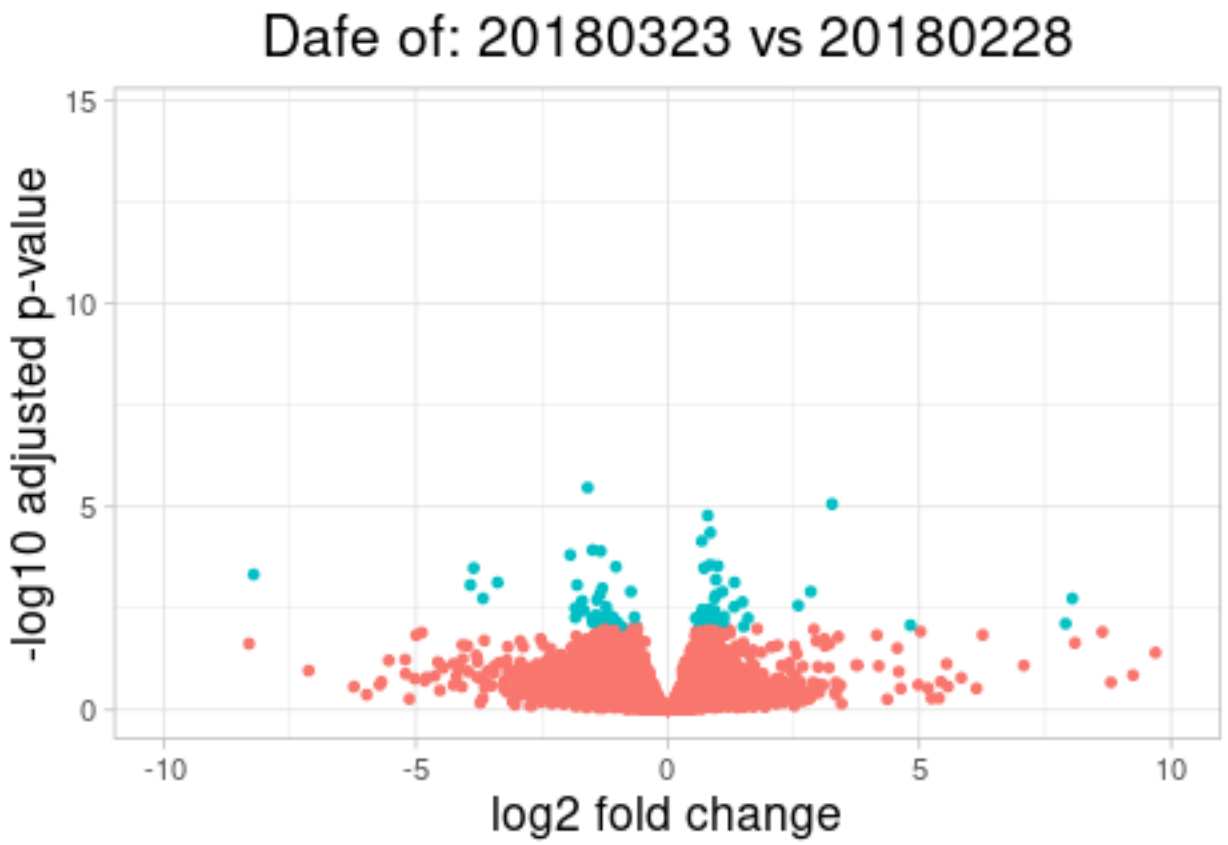
```r
# Add a column for significant genes
resER_tb <- resER_tb %>% mutate(threshold = padj < 0.01)

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



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

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

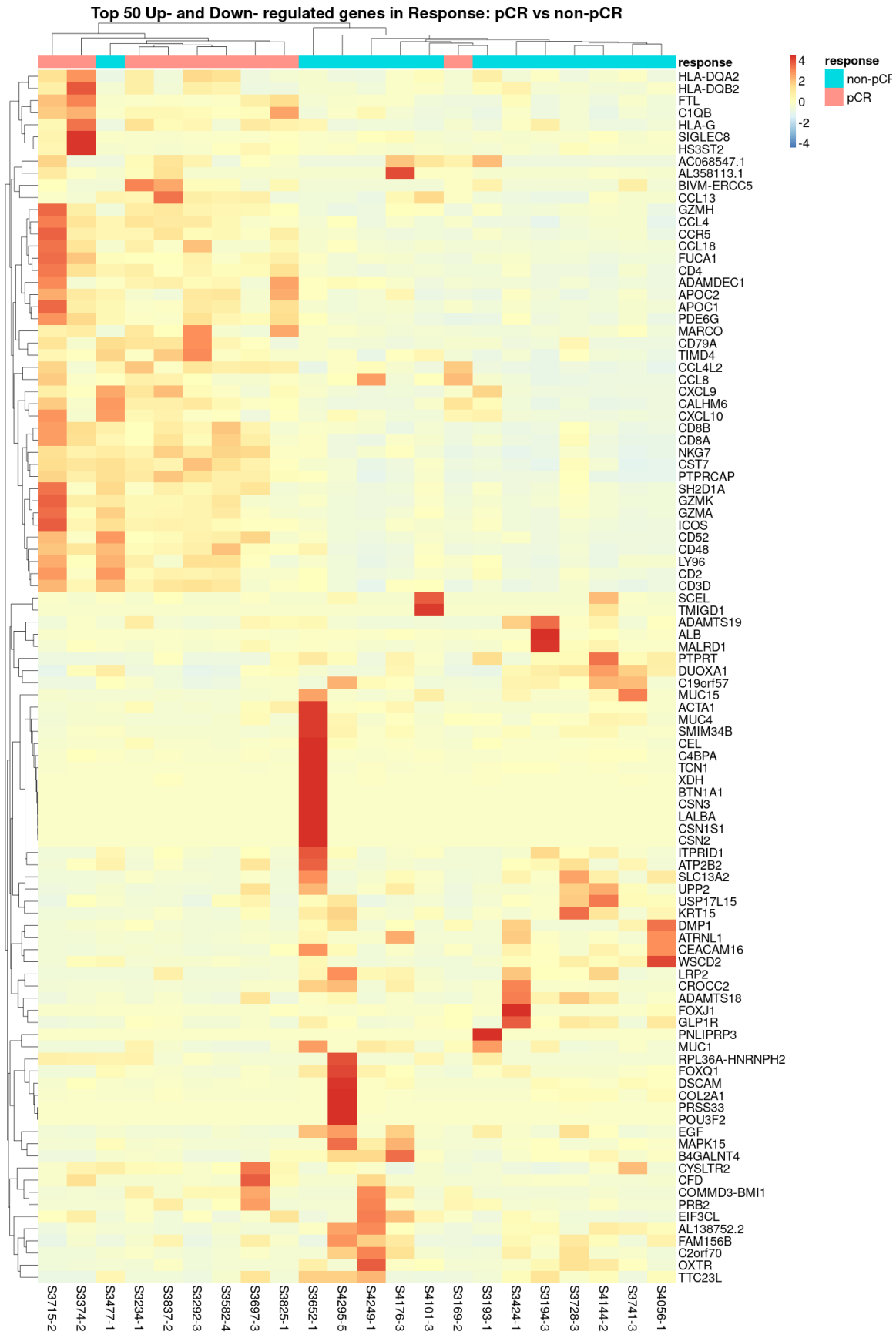


```r
# Add a column for significant genes
resDO_tb <- resDO_tb %>% mutate(threshold = padj < 0.01)

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



Heatmaps

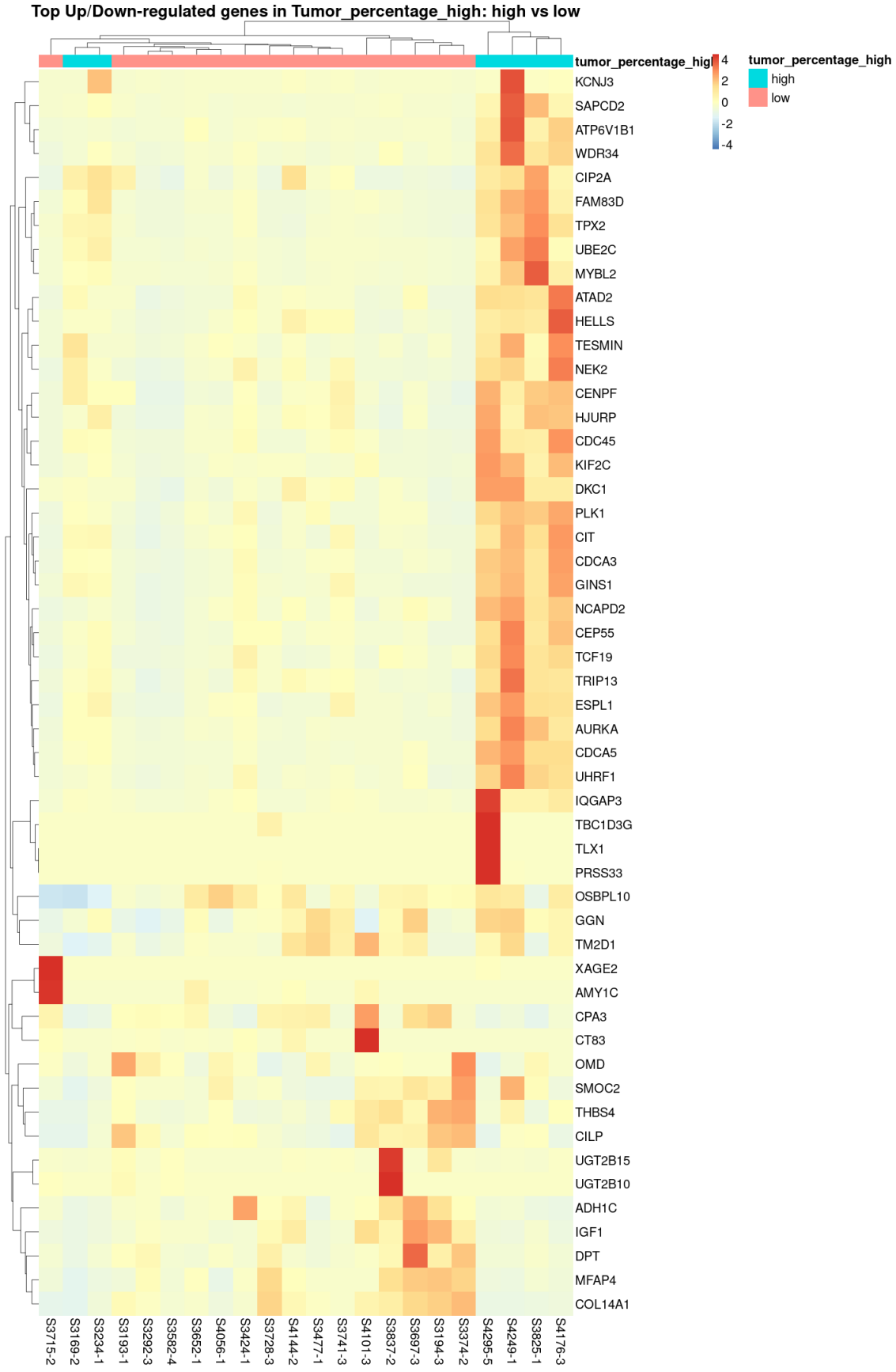
```
# Create a matrix of normalized expression
sig_up <- resResponse_tb_significant %>% arrange(-log2FoldChange) %>% head(50) %>% pull(gene)
sig_down <- resResponse_tb_significant %>% arrange(log2FoldChange) %>% head(50) %>% pull(gene)
sig <- c(sig_up, sig_down)

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

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

plotmat$ensembl_gene_id <- NULL

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

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

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


Top 50 Up- and Down- regulated genes in Response: pCR vs non-pCR
response
HLA-DQA2
HLA-DQB2
FTL
C1QB
HLA-G
SIGLEC8
HS3ST2
AC068547.1
AL358113.1
BIVM-ERCC5
CCL13
GZMH
CCL4
CCR5
CCL18
FUCA1
CD4
ADAMDEC1
APOC2
APOC1
PDE6G
MARCO
CD79A
TIMD4
CCL4L2
CCL8
CXCL9
CALHM6
CXCL10
CD8B
CD8A
NKG7
CST7
PTPRCAP
SH2D1A
GZMK
GZMA
ICOS
CD52
CD48
LY96
CD2
CD3D
SCEL
TMIGD1
ADAMTS19
ALB
MALRD1
PTPRT
DUOXA1
C19orf57
MUC15
ACTA1
MUC4
SMIM34B
CEL
C4BPA
TCN1
XDH
BTN1A1
CSN3
LALBA
CSN1S1
CSN2
ITPRID1
ATP2B2
SLC13A2
UPP2
USP17L15
KRT15
DMP1
ATRNL1
CEACAM16
WSCD2
LRP2
CROCC2
ADAMTS18
FOXJ1
GLP1R
PNLIPRP3
MUC1
RPL36A-HNRNPH2
FOXQ1
DSCAM
COL2A1
PRSS33
POU3F2
EGF
MAPK15
B4GALNT4
CYSLTR2
CFD
COMMD3-BMI1
PRB2
EIF3CL
AL138752.2
FAM156B
C2orf70
OXTR
TTC23L
S3715-2
S3374-2
S3477-1
S3234-1
S3837-2
S3292-3
S3582-4
S3697-3
S3825-1
S3652-1
S4295-5
S4249-1
S4176-3
S4101-3
S3169-2
S3193-1
S3424-1
S3194-3
S3728-3
S4144-2
S3741-3
S4056-1
4
2
0
-2
-4
response
non-pCR
pCR

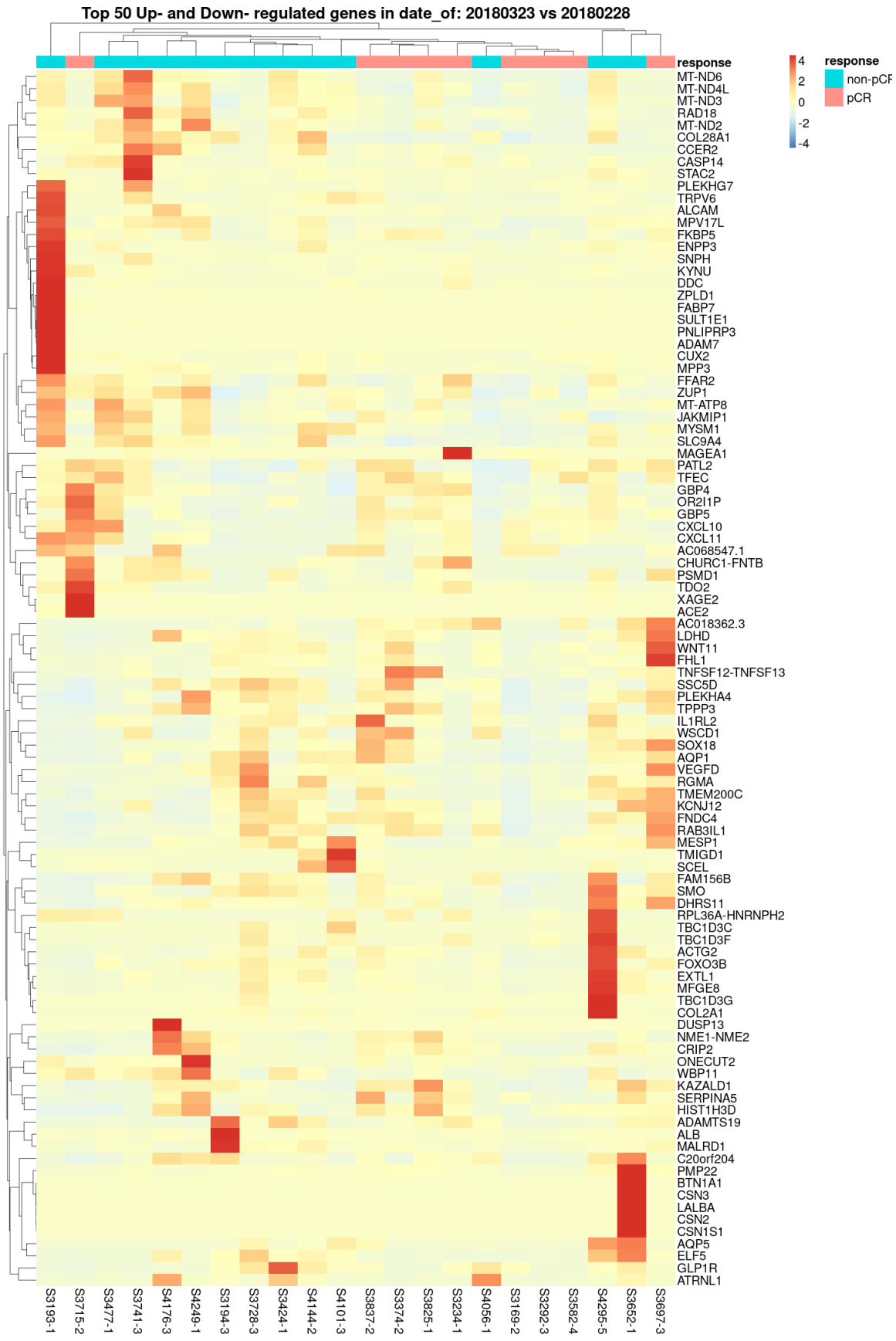
```r
# Create a matrix of normalized expression
sig_up <- resER_tb_significant %>% arrange(-log2FoldChange) %>% head(50) %>% pull(gene)
sig_down <- resER_tb_significant %>% arrange(log2FoldChange) %>% head(50) %>% pull(gene)
sig <- c(sig_up, sig_down)

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
er
Negativ
Positive
4
2
0
-2
-4
BAALC
SOX18
HRNR
ADAMTS9
GPC2
GOLGA8N
CMTM7
DLL1
NLGN2
TTYH2
CHST3
FAM171A1
DTX1
L3MBTL4
EPHB1
HOXA3
OTOF
SPNS2
WNT6
PRKAG3
TLX1
POU3F2
TBC1D3G
OCA2
HAPLN3
EXTL1
MFGE8
MAPK4
FOXC1
ACTG2
STOX2
PROX1
RYR2
STON1-GTF2A1L
AOC2
DLGAP2
CRYAB
MAML2
ITGA9
PITPNM2
NTF3
COL27A1
TMIGD1
COL11A2
C4BPA
LALBA
CSN2
CSN1S1
TNFSF12-TNFSF13
CILP
OMD
AC068547.1
XAGE2
LRP1B
ALCAM
AR
SLC26A3
PNLIPRP3
ADAM7
SDF2
ARFGEF3
JUP
SLC19A2
FSIP1
ALDH3B2
CLEC3A
SOWAHA
TOM1L1
SHISA2
NPNT
NLK
TK1
TMEM97
DUSP4
NKAIN1
NBPF4
PARD6B
GRM4
HID1
ANKRD30B
ESR1
ARMT1
GATA3
PRLR
HOXC10
H2AFJ
LRFN2
COX6C
NAT1
GPRC5A
TOP2A
KRT18
KRT8
XBP1
FBP1
HOXC11
MSX2
HOXC9
SLC39A6
PSMD5
S4295-5
S4249-1
S3424-1
S4056-1
S3194-3
S3169-2
S3234-1
S3374-2
S3477-1
S3741-3
S4101-3
S3715-2
S3292-3
S3582-4
S3652-1
S4144-2
S3697-3
S3728-3
S3837-2
S3193-1
S3825-1
S4176-3

```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
tumor_percentage_high
4
2
0
-2
-4
tumor_percentage_high
high
low
KCNJ3
SAPCD2
ATP6V1B1
WDR34
CIP2A
FAM83D
TPX2
UBE2C
MYBL2
ATAD2
HELLS
TESMIN
NEK2
CENPF
HJURP
CDC45
KIF2C
DKC1
PLK1
CIT
CDCA3
GINS1
NCAPD2
CEP55
TCF19
TRIP13
ESPL1
AURKA
CDCA5
UHRF1
IQGAP3
TBC1D3G
TLX1
PRSS33
OSBPL10
GGN
TM2D1
XAGE2
AMY1C
CPA3
CT83
OMD
SMOC2
THBS4
CILP
UGT2B15
UGT2B10
ADH1C
IGF1
DPT
MFAP4
COL14A1
S3715-2
S3169-2
S3234-1
S3193-1
S3292-3
S3582-4
S3652-1
S4056-1
S3424-1
S3728-3
S4144-2
S3477-1
S3741-3
S4101-3
S3837-2
S3697-3
S3194-3
S3374-2
S4295-5
S4249-1
S3825-1
S4176-3

```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
non-pCR
pCR
4
2
0
-2
-4
MT-ND6
MT-ND4L
MT-ND3
RAD18
MT-ND2
COL28A1
CCER2
CASP14
STAC2
PLEKHG7
TRPV6
ALCAM
MPV17L
FKBP5
ENPP3
SNPH
KYNU
DDC
ZPLD1
FABP7
SULT1E1
PNLIPRP3
ADAM7
CUX2
MPP3
FFAR2
ZUP1
MT-ATP8
JAKMIP1
MYSM1
SLC9A4
MAGEA1
PATL2
TFEC
GBP4
OR2I1P
GBP5
CXCL10
CXCL11
AC068547.1
CHURC1-FNTB
PSMD1
TDO2
XAGE2
ACE2
AC018362.3
LDHD
WNT11
FHL1
TNFSF12-TNFSF13
SSC5D
PLEKHA4
TPPP3
IL1RL2
WSCD1
SOX18
AQP1
VEGFD
RGMA
TMEM200C
KCNJ12
FNDC4
RAB3IL1
MESP1
TMIGD1
SCEL
FAM156B
SMO
DHRS11
RPL36A-HNRNPH2
TBC1D3C
TBC1D3F
ACTG2
FOXO3B
EXTL1
MFGE8
TBC1D3G
COL2A1
DUSP13
NME1-NME2
CRIP2
ONECUT2
WBP11
KAZALD1
SERPINA5
HIST1H3D
ADAMTS19
ALB
MALRD1
C20orf204
PMP22
BTN1A1
CSN3
LALBA
CSN2
CSN1S1
AQP5
ELF5
GLP1R
ATRNL1
S3193-1
S3715-2
S3477-1
S3741-3
S4176-3
S4249-1
S3194-3
S3728-3
S3424-1
S4144-2
S4101-3
S3837-2
S3374-2
S3825-1
S3234-1
S4056-1
S3169-2
S3292-3
S3582-4
S4295-5
S3652-1
S3697-3

Functional analysis

Biological Process (BP)

```
bg_genes <- rownames(resResponse)

## Run GO enrichment analysis
compgo_file <- "data/day8.compgo.up.RDS"
if (file.exists(compgo_file)){
    compGO <- readRDS(compgo_file)
}else{
    compGO <- enrichGO(gene = sigResponse_up,
                   universe = bg_genes,
                   keyType = "ENSEMBL",
                   OrgDb = "org.Hs.eg.db",
                   ont = "BP",
                   qvalueCutoff  = 0.05,
                   pAdjustMethod = "BH",
                   readable = TRUE)
    saveRDS(compGO, compgo_file)
}

dotplot(compGO,
        showCategory = 20,
        title = "GO (Biological Process) Enrichment \n Analysis for UP in pCR)",
        label_format = 20,
        font.size = 10)
```


```
## Output results from GO analysis to a table

print("UP")
```

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

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

```
## [1] 458
```

```
write_csv(results_up, "tables/T20.day8.GO_BP_UP.csv")
```

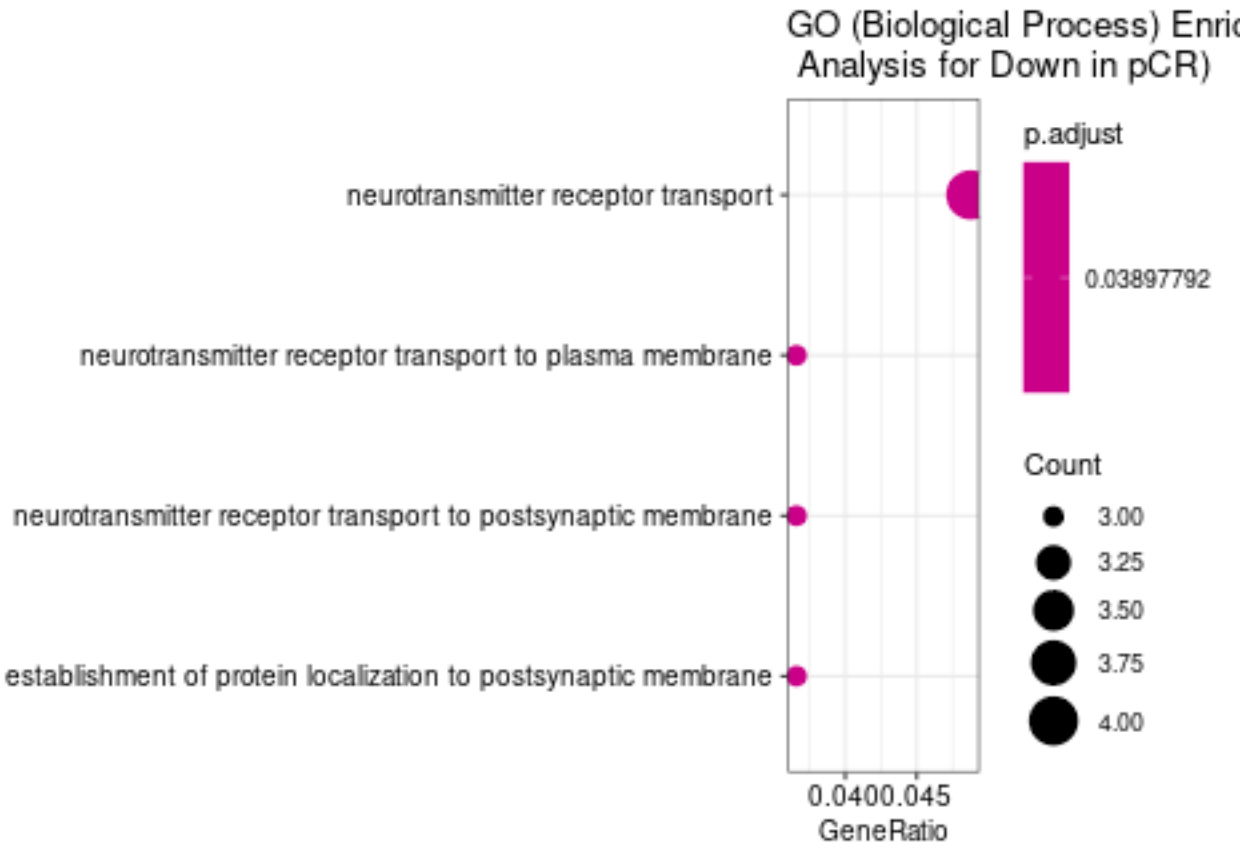
```
compgo_file <- "data/day8.compgo.down.RDS"
if (file.exists(compgo_file)){
    compGO <- readRDS(compgo_file)
}else{
    compGO <- enrichGO(gene = sigResponse_down,
                    universe = bg_genes,
                    keyType = "ENSEMBL",
                    OrgDb = "org.Hs.eg.db",
                    ont = "BP",
                    qvalueCutoff  = 0.05,
                    pAdjustMethod = "BH",
                    readable = TRUE)
    saveRDS(compGO, compgo_file)
}


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

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

```
nrow(results_down)
```

```
## [1] 4
```

```
write_csv(results_down, "tables/T21.day8.GO_BP_DOWN.csv")
dotplot(compGO,
        showCategory = 20,
        title = "GO (Biological Process) Enrichment \n Analysis for Down in pCR)",
        label_format = 20,
        font.size = 10)
```


GO (Biological Process) Enric
Analysis for Down in pCR)
neurotransmitter receptor transport
neurotransmitter receptor transport to plasma membrane
neurotransmitter receptor transport to postsynaptic membrane
establishment of protein localization to postsynaptic membrane
0.040
0.045
GeneRatio
p.adjust
0.03897792
Count
3.00
3.25
3.50
3.75
4.00

Molecular Function (MF)

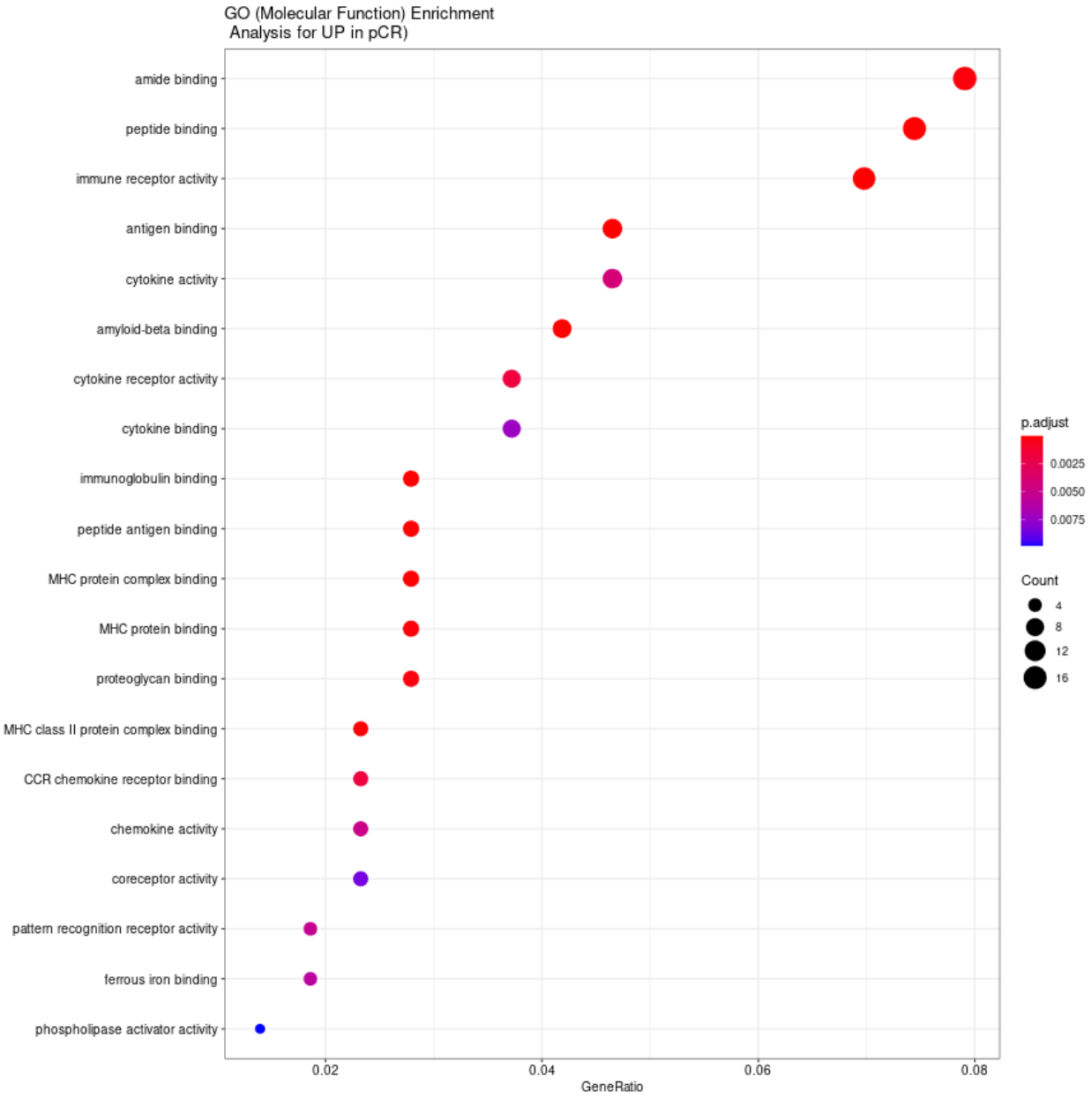
```
bg_genes <- rownames(resResponse)

## Run GO enrichment analysis
compgo_file <- "data/day8.compgo.up.mf.RDS"
if (file.exists(compgo_file)){
    compGO <- readRDS(compgo_file)
}else{
    compGO <- enrichGO(gene = sigResponse_up,
                    universe = bg_genes,
                    keyType = "ENSEMBL",
                    OrgDb = "org.Hs.eg.db",
                    ont = "MF",
                    qvalueCutoff  = 0.05,
                    pAdjustMethod = "BH",
                    readable = TRUE)
    saveRDS(compGO, compgo_file)
}
# 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)
nrow(results_up)
```

```
## [1] 28
```

```
write_csv(results_up, "tables/T22.day8.GO_MF_UP.csv")

dotplot(compGO,
        showCategory = 20,
        title = "GO (Molecular Function) Enrichment \n Analysis for UP in pCR)",
        label_format = 20,
        font.size = 10)
```


GO (Molecular Function) Enrichment
Analysis for UP in pCR)
amide binding
peptide binding
immune receptor activity
antigen binding
cytokine activity
amyloid-beta binding
cytokine receptor activity
cytokine binding
immunoglobulin binding
peptide antigen binding
MHC protein complex binding
MHC protein binding
proteoglycan binding
MHC class II protein complex binding
CCR chemokine receptor binding
chemokine activity
coreceptor activity
pattern recognition receptor activity
ferrous iron binding
phospholipase activator activity
0.02
0.04
0.06
0.08
GeneRatio
p.adjust
0.0025
0.0050
0.0075
Count
4
8
12
16

```
compgo_file <- "data/day8.compgo.down.mf.RDS"
if (file.exists(compgo_file)){
    compGO <- readRDS(compgo_file)
}else{
  compGO <- enrichGO(gene = sigResponse_down,
                     universe = bg_genes,
                     keyType = "ENSEMBL",
                     OrgDb = "org.Hs.eg.db",
                     ont = "MF",
                     qvalueCutoff  = 0.05,
                     pAdjustMethod = "BH",
                     readable = TRUE)
   saveRDS(compGO, compgo_file)
}

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

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

```
nrow(results_down)
```

```
## [1] 0
```

```
write_csv(results_up, "tables/T22.day8.GO_MF_DOWN.csv")

if (nrow(results_down) > 0){
  dotplot(compGO,
        showCategory = 20,
        title = "GO (Molecular Function) Enrichment \n Analysis for DOWN in pCR)",
        label_format = 20,
        font.size = 10)
}
```

Cellular Compartment (CC)

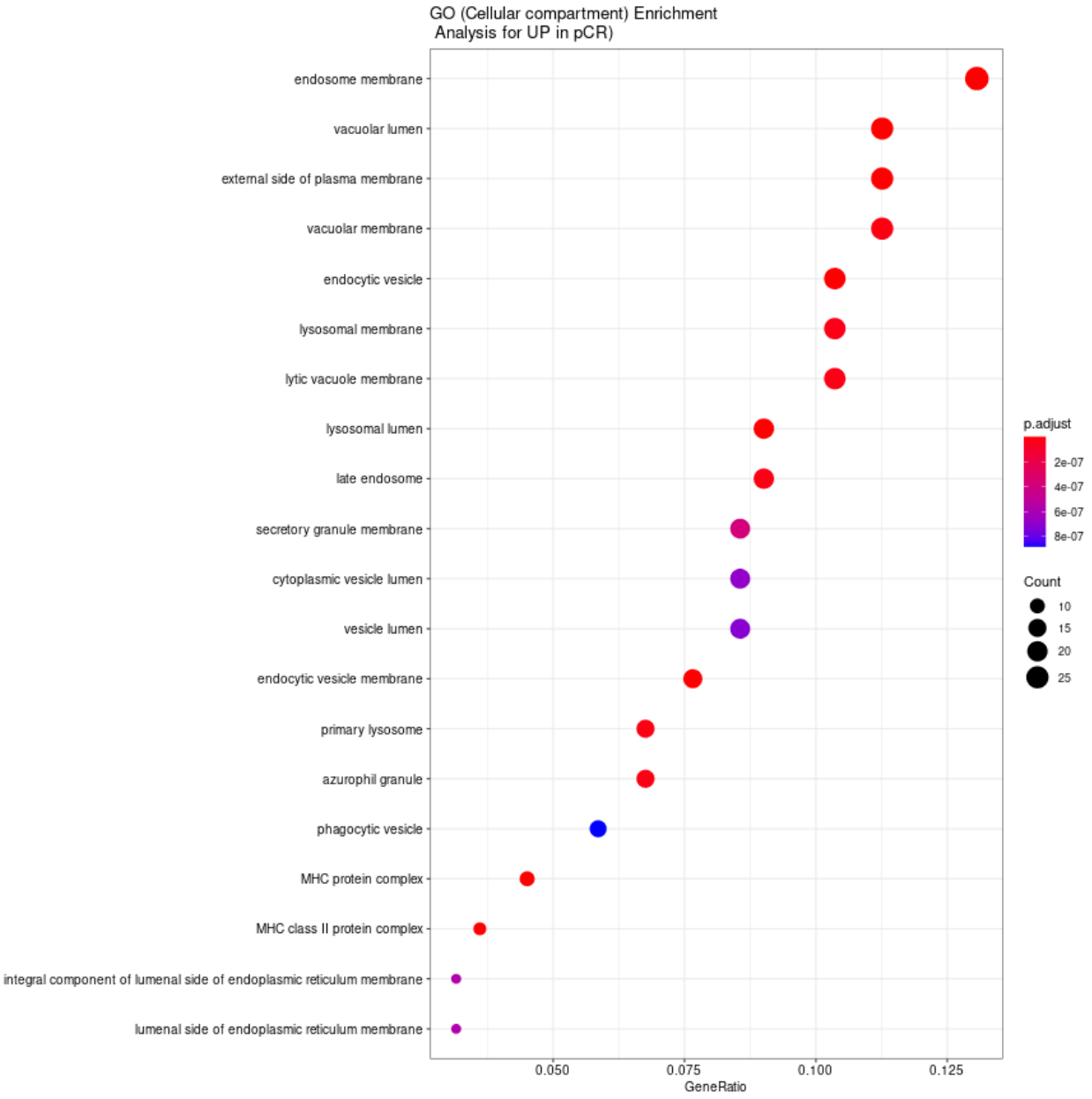
```
bg_genes <- rownames(resResponse)

## Run GO enrichment analysis
compgo_file <- "data/day8.compgo.up.cc.RDS"
if (file.exists(compgo_file)){
    compGO <- readRDS(compgo_file)
}else{
  compGO <- enrichGO(gene = sigResponse_up,
                     universe = bg_genes,
                     keyType = "ENSEMBL",
                     OrgDb = "org.Hs.eg.db",
                     ont = "CC",
                     qvalueCutoff  = 0.05,
                     pAdjustMethod = "BH",
                     readable = TRUE)
  saveRDS(compGO, compgo_file)
}

# 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)
nrow(results_up)
```

```
## [1] 71
```

```
write_csv(results_up, "tables/T23.day8.GO_CC_UP.csv")


dotplot(compGO,
        showCategory = 20,
        title = "GO (Cellular compartment) Enrichment \n Analysis for UP in pCR)",
        label_format = 20,
        font.size = 10)
```


GO (Cellular compartment) Enrichment
Analysis for UP in pCR)
endosome membrane
vacuolar lumen
external side of plasma membrane
vacuolar membrane
endocytic vesicle
lysosomal membrane
lytic vacuole membrane
lysosomal lumen
late endosome
secretory granule membrane
cytoplasmic vesicle lumen
vesicle lumen
endocytic vesicle membrane
primary lysosome
azurophil granule
phagocytic vesicle
MHC protein complex
MHC class II protein complex
integral component of lumenal side of endoplasmic reticulum membrane
lumenal side of endoplasmic reticulum membrane
0.050
0.075
0.100
0.125
GeneRatio
p.adjust
2e-07
4e-07
6e-07
8e-07
Count
10
15
20
25

```
compgo_file <- "data/day8.compgo.down.cc.RDS"
if (file.exists(compgo_file)){
    compGO <- readRDS(compgo_file)
}else{
    compGO <- enrichGO(gene = sigResponse_down,
                    universe = bg_genes,
                    keyType = "ENSEMBL",
                    OrgDb = "org.Hs.eg.db",
                    ont = "CC",
                    qvalueCutoff  = 0.05,
                    pAdjustMethod = "BH",
                    readable = TRUE)
    saveRDS(compGO, compgo_file)
}

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

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

```
nrow(results_down)
```

```
## [1] 0
```

```
write_csv(results_up, "tables/T24.day8.GO_CC_DOWN.csv")

if (nrow(results_down) > 0){
dotplot(compGO,
        showCategory = 20,
        title = "GO (Cellular compartment) Enrichment \n Analysis for DOWN in pCR)",
        label_format = 20,
        font.size = 10)
}
```

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] clusterProfiler_3.18.1      org.Hs.eg.db_3.12.0
##  [3] ensembldb_2.14.1            AnnotationFilter_1.14.0
##  [5] GenomicFeatures_1.42.3      AnnotationDbi_1.52.0
##  [7] AnnotationHub_2.22.1        BiocFileCache_1.14.0
##  [9] dbplyr_2.1.1                knitr_1.30
## [11] ggrepel_0.9.1               tximport_1.18.0
## [13] DEGreport_1.26.0            pheatmap_1.0.12
## [15] RColorBrewer_1.1-2          forcats_0.5.1
## [17] stringr_1.4.0               dplyr_1.0.5
## [19] purrr_0.3.4                 readr_1.4.0
## [21] tidyr_1.1.3                 tibble_3.1.1
## [23] ggplot2_3.3.3               tidyverse_1.3.1
## [25] DESeq2_1.30.1               SummarizedExperiment_1.20.0
## [27] Biobase_2.50.0              MatrixGenerics_1.2.1
## [29] matrixStats_0.58.0          GenomicRanges_1.42.0
## [31] GenomeInfoDb_1.26.7         IRanges_2.24.1
## [33] S4Vectors_0.28.1            BiocGenerics_0.36.1
##
## loaded via a namespace (and not attached):
##   [1] utf8_1.2.1                    tidyselect_1.1.0
##   [3] RSQLite_2.2.7                 grid_4.0.3
##   [5] BiocParallel_1.24.1           scatterpie_0.1.6
##   [7] munsell_0.5.0                 withr_2.4.2
##   [9] colorspace_2.0-0              GOSemSim_2.16.1
##  [11] rstudioapi_0.13               DOSE_3.16.0
##  [13] labeling_0.4.2                lasso2_1.2-21.1
##  [15] GenomeInfoDbData_1.2.4        polyclip_1.10-0
##  [17] mnormt_2.0.2                  farver_2.1.0
##  [19] bit64_4.0.5                   downloader_0.4
##  [21] vctrs_0.3.7                   generics_0.1.0
##  [23] xfun_0.19                     R6_2.5.0
##  [25] graphlayouts_0.7.1            clue_0.3-59
##  [27] locfit_1.5-9.4                bitops_1.0-7
##  [29] cachem_1.0.4                  reshape_0.8.8
##  [31] fgsea_1.16.0                  DelayedArray_0.16.3
##  [33] assertthat_0.2.1              promises_1.2.0.1
##  [35] scales_1.1.1                  ggraph_2.0.5
##  [37] enrichplot_1.10.2             gtable_0.3.0
##  [39] Cairo_1.5-12.2                conquer_1.0.2
##  [41] tidygraph_1.2.0               MatrixModels_0.5-0
##  [43] rlang_0.4.10                  genefilter_1.72.1
##  [45] GlobalOptions_0.1.2           splines_4.0.3
##  [47] rtracklayer_1.50.0            lazyeval_0.2.2
##  [49] broom_0.7.6                   BiocManager_1.30.12
##  [51] yaml_2.2.1                    reshape2_1.4.4
##  [53] modelr_0.1.8                  backports_1.2.1
##  [55] httpuv_1.6.0                  qvalue_2.22.0
##  [57] tools_4.0.3                   psych_2.1.3
##  [59] logging_0.10-108              ellipsis_0.3.1
##  [61] ggdendro_0.1.22               Rcpp_1.0.6
```

```
##  [63] plyr_1.8.6                   progress_1.2.2
##  [65] zlibbioc_1.36.0              RCurl_1.98-1.3
##  [67] prettyunits_1.1.1            openssl_1.4.3
##  [69] viridis_0.6.0                GetoptLong_1.0.5
##  [71] cowplot_1.1.1                haven_2.4.1
##  [73] cluster_2.1.0                fs_1.5.0
##  [75] magrittr_2.0.1               data.table_1.14.0
##  [77] DO.db_2.9                    SparseM_1.81
##  [79] circlize_0.4.12              reprex_2.0.0
##  [81] tmvnsim_1.0-2                ProtGenerics_1.22.0
##  [83] hms_1.0.0                    mime_0.9
##  [85] evaluate_0.14                xtable_1.8-4
##  [87] XML_3.99-0.6                 readxl_1.3.1
##  [89] gridExtra_2.3                shape_1.4.5
##  [91] compiler_4.0.3               biomaRt_2.46.3
##  [93] shadowtext_0.0.8             crayon_1.4.1
##  [95] htmltools_0.5.1.1            later_1.2.0
##  [97] geneplotter_1.68.0           lubridate_1.7.10
##  [99] DBI_1.1.1                    tweenr_1.0.2
## [101] ComplexHeatmap_2.6.2         MASS_7.3-53
## [103] rappdirs_0.3.3               Matrix_1.2-18
## [105] cli_2.5.0                    igraph_1.2.6
## [107] pkgconfig_2.0.3              rvcheck_0.1.8
## [109] GenomicAlignments_1.26.0     xml2_1.3.2
## [111] annotate_1.68.0              XVector_0.30.0
## [113] rvest_1.0.0                  digest_0.6.27
## [115] ConsensusClusterPlus_1.54.0  Biostrings_2.58.0
## [117] rmarkdown_2.5                cellranger_1.1.0
## [119] fastmatch_1.1-0              edgeR_3.32.1
## [121] curl_4.3                     quantreg_5.85
## [123] shiny_1.6.0                  Rsamtools_2.6.0
## [125] rjson_0.2.20                 lifecycle_1.0.0
## [127] nlme_3.1-149                 jsonlite_1.7.2
## [129] viridisLite_0.4.0            askpass_1.1
## [131] limma_3.46.0                 fansi_0.4.2
## [133] pillar_1.6.0                 lattice_0.20-41
## [135] Nozzle.R1_1.1-1              fastmap_1.1.0
## [137] httr_1.4.2                   survival_3.2-7
## [139] GO.db_3.12.1                 interactiveDisplayBase_1.28.0
## [141] glue_1.4.2                   png_0.1-7
## [143] BiocVersion_3.12.0           bit_4.0.4
## [145] ggforce_0.3.3                stringi_1.5.3
## [147] blob_1.2.1                   memoise_2.0.0
```