RNA-seq course- week1

Serhiy Naumenko

2023-09-26

Contents

Overview

- Schlotawa data reanalysis for the RNA-seq course

Load Counts

```
# raw counts downloaded from GEO
# https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE205555

counts_csv <- "tables/counts.csv"
counts_tpm_csv <- "tables/counts_tpm.csv"
if (file.exists(counts_csv)){
    counts <- read_csv(counts_csv) %>% column_to_rownames("ensembl_gene_id")
    counts_tpm <- read_csv(counts_tpm_csv)
}else{
    files <- fs::dir_ls(path = "../data/input_geo", glob = "*exonCounts.txt")
    counts <- readr::read_tsv(files, id = "path", col_names = c("ensembl_gene_id", "raw_counts"))
    df_split <- str_split_fixed(counts$path, "_", 13) %>% as.data.frame()
    counts$sample <- df_split$V2
    counts <- counts %>%
        mutate(sample = str_replace(sample, "geo/", "")) %>%
        dplyr::relocate(sample) %>% dplyr::select(-path) %>%
        pivot_wider(names_from = "sample", values_from = "raw_counts")


    gene_length <- read_tsv("tables/GC_lengths.tsv") %>% arrange(ensembl_gene_id)
    counts <- counts %>% arrange(ensembl_gene_id)
    gene_ids <- intersect(counts$ensembl_gene_id, gene_length$ensembl_gene_id)

    v_len <- gene_length %>% dplyr::filter(ensembl_gene_id %in% gene_ids)
    counts <- counts %>% dplyr::filter(ensembl_gene_id %in% gene_ids)

    write_csv(counts, counts_csv)
    counts <- counts %>% column_to_rownames("ensembl_gene_id")

    x <- counts / v_len$Length
    counts_tpm <- t(t(x) * 1e6 / colSums(x)) %>% as.data.frame() %>% round(2) %>%
      rownames_to_column("ensembl_gene_id") %>% write_csv(counts_tpm_csv)
}
```

Load metadata

```
# Load the data and metadata
metadata <- read_csv("tables/metadata.csv") %>% column_to_rownames(var = "sample_id")

protein_coding_genes <- read_csv("tables/ensembl_w_description.protein_coding.csv")
```

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

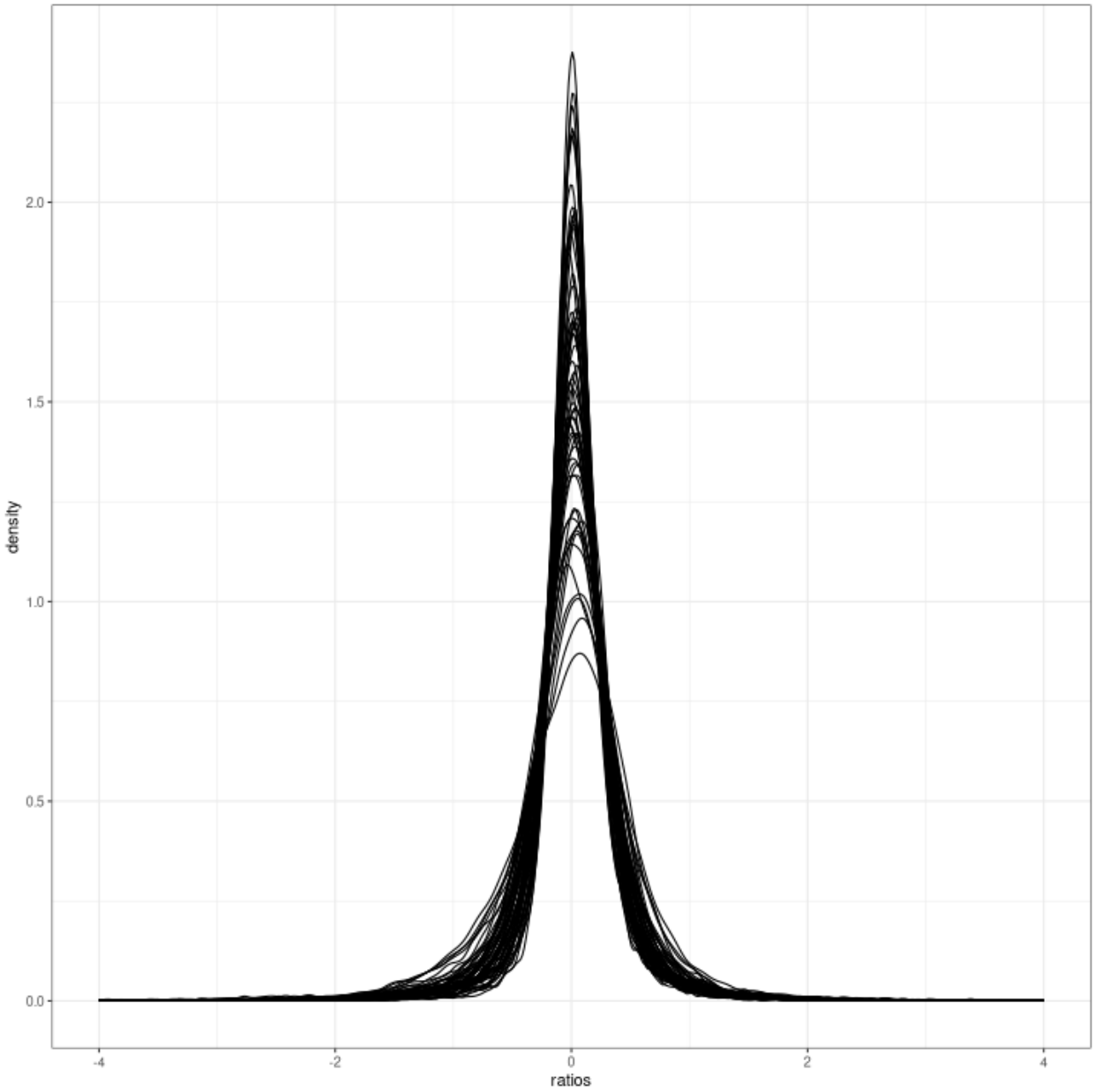
Run DEseq2 Wald test

Here we subset protein coding genes.

DEGreport QC

Size factor QC

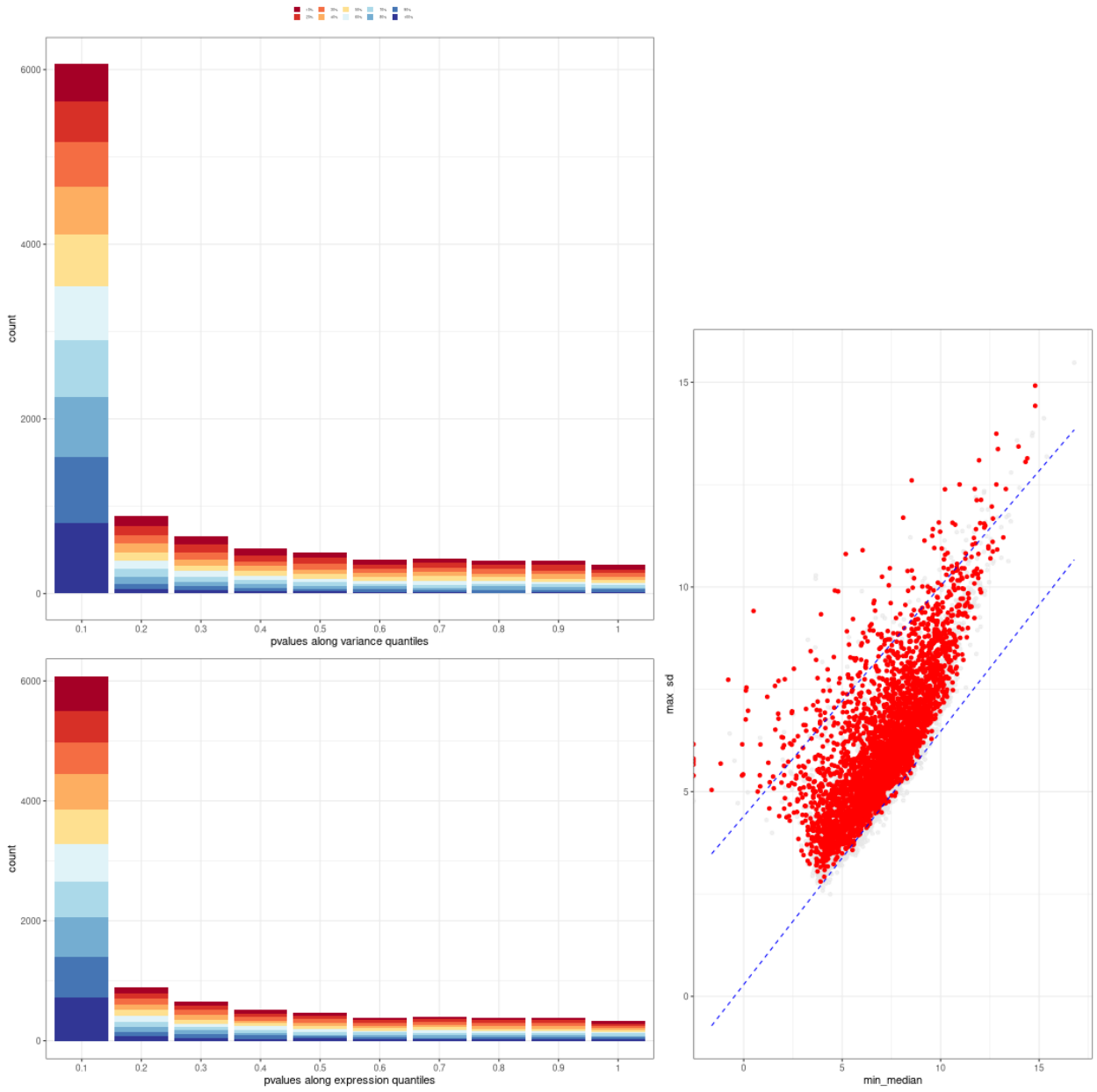
```
counts <- counts(dds, normalized = TRUE)
design <- as.data.frame(colData(dds))
degCheckFactors(counts)
```



Mean-Variance QC plots

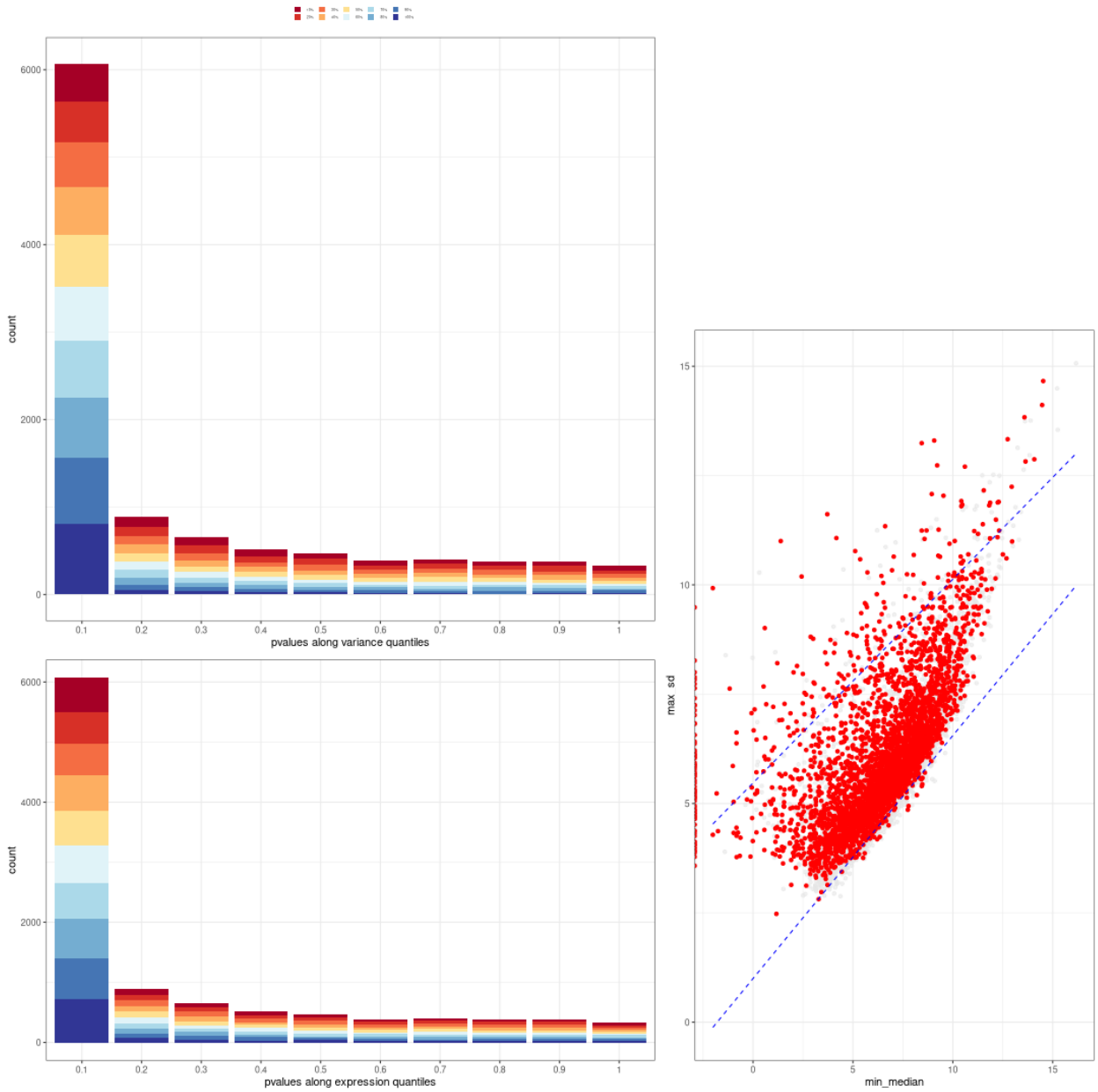
treatment

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



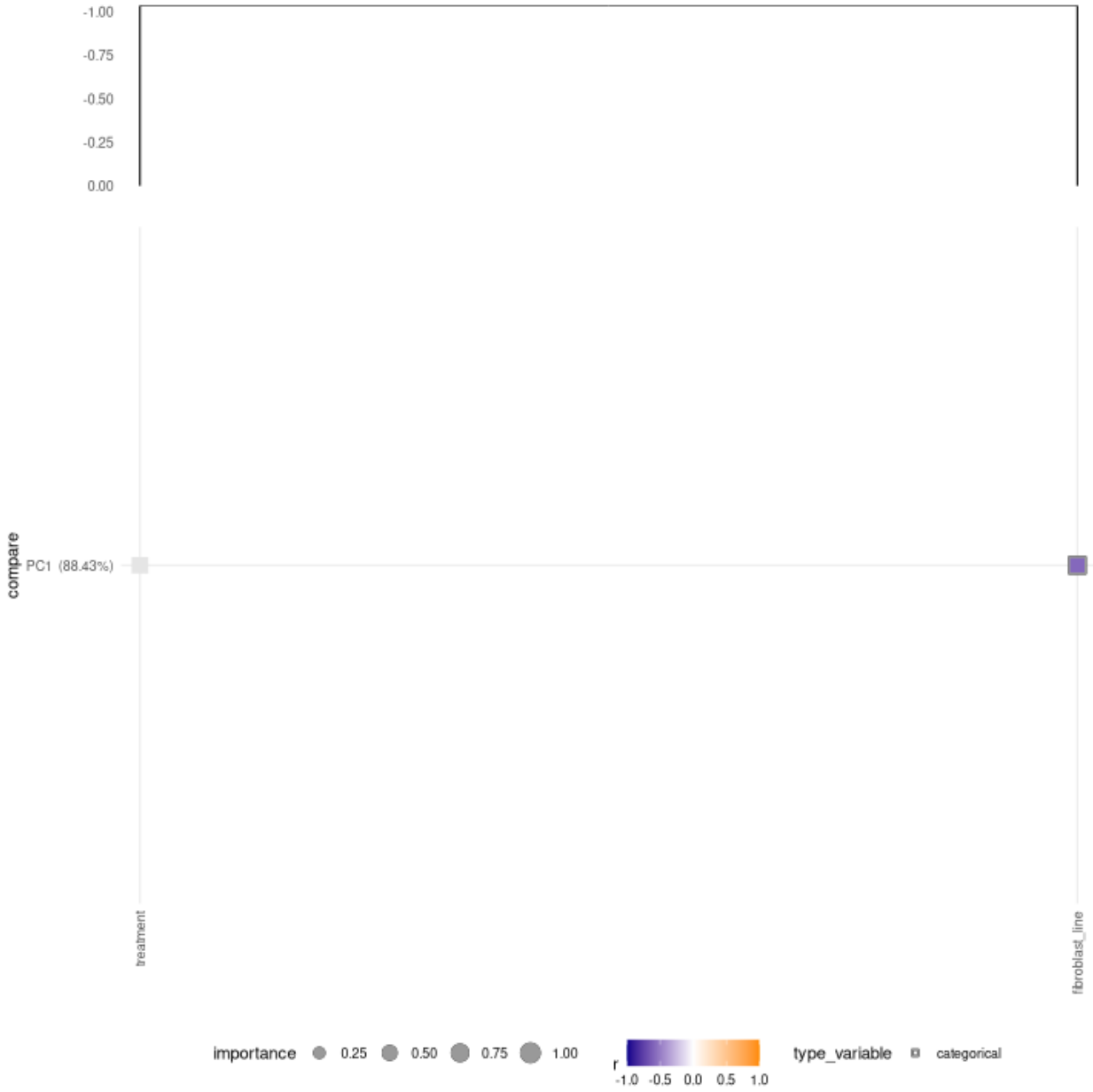
fibroblast_line

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



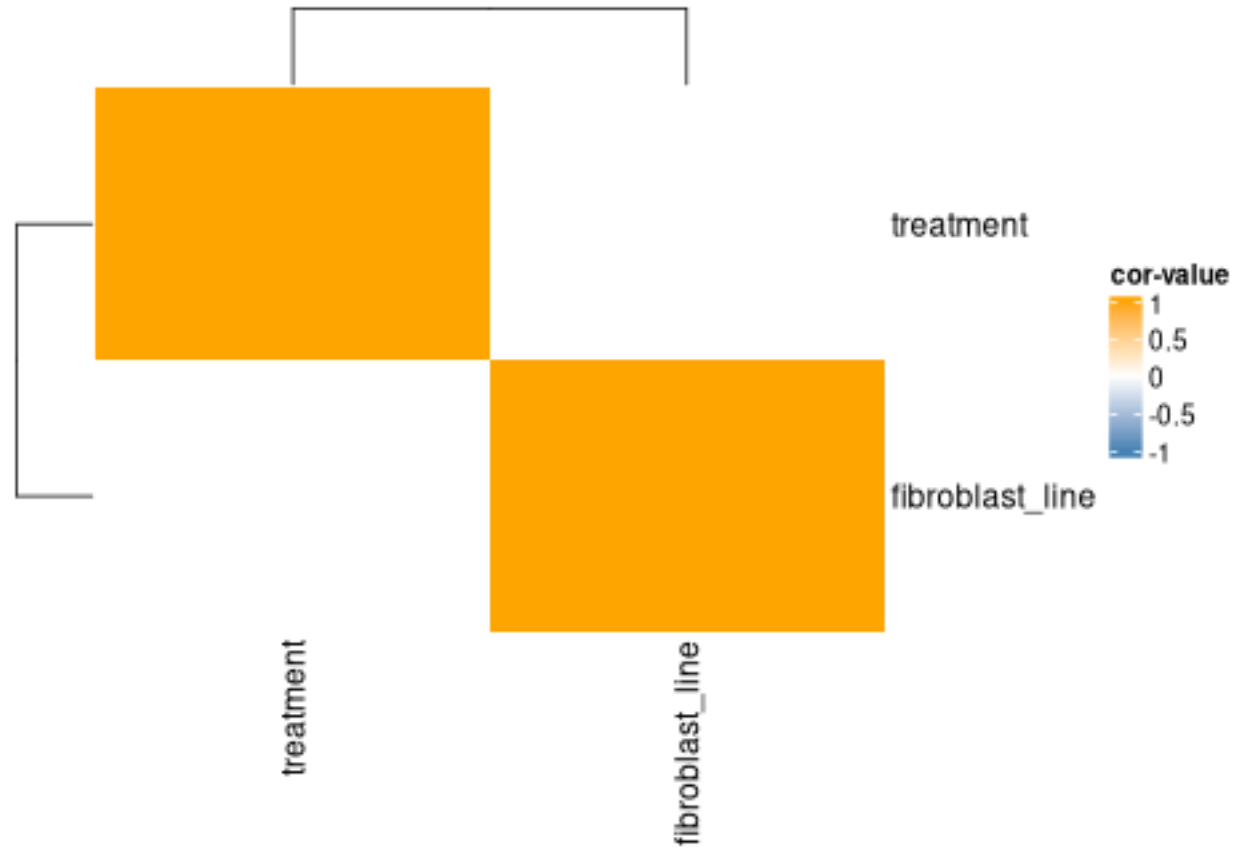
Covariates effect on count data

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



Covariates correlation with metrics

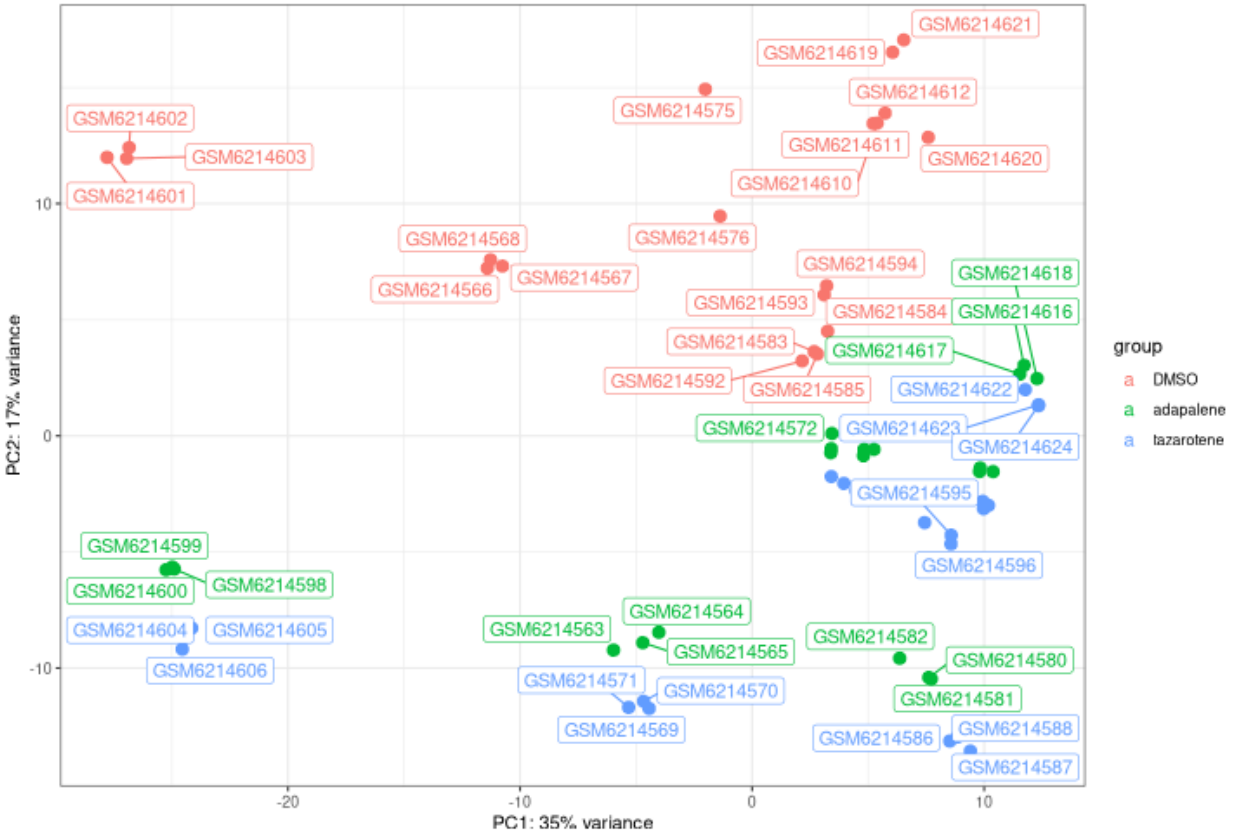
```
cor <- degCorCov(mdata)
```



Sample-level QC analysis

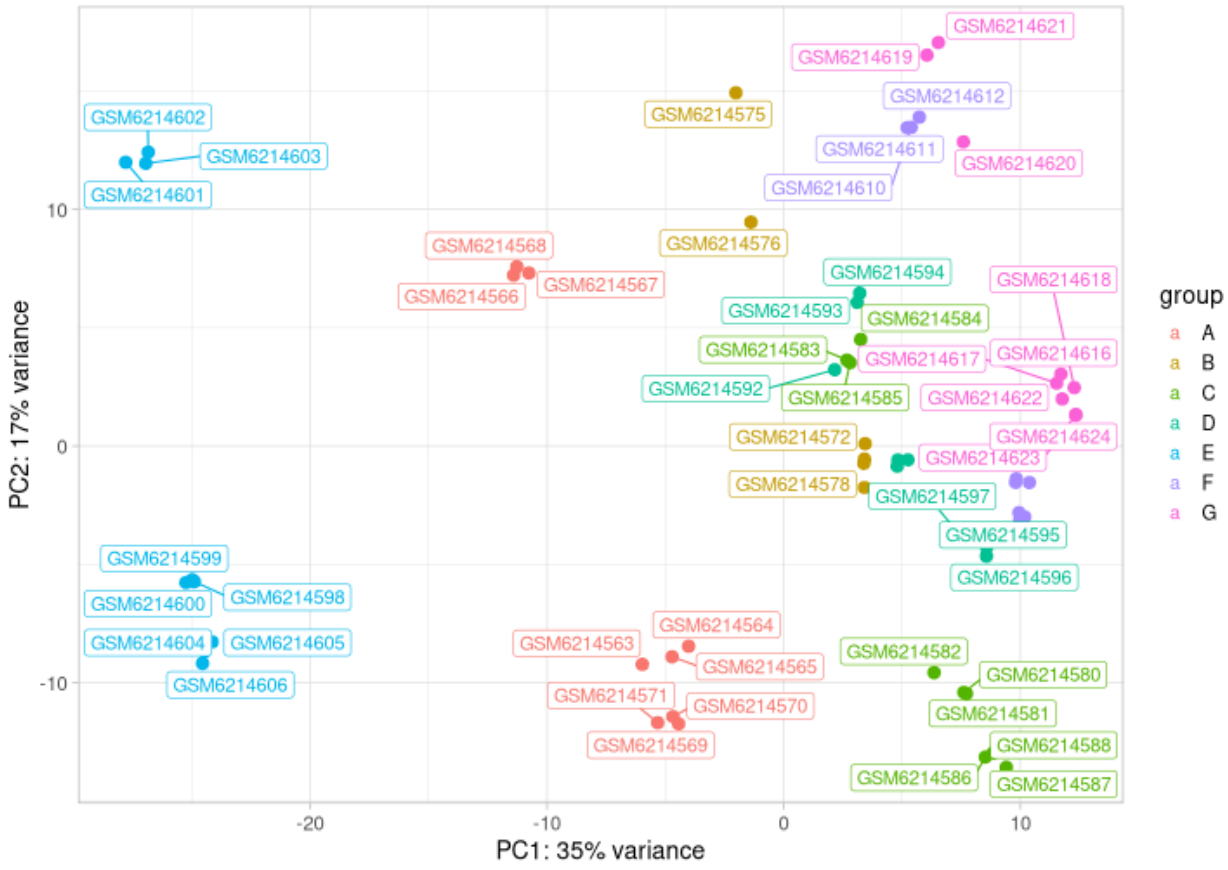
PCA - treatment

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


GSM6214621
GSM6214619
GSM6214575
GSM6214612
GSM6214602
GSM6214611
GSM6214620
GSM6214603
GSM6214610
GSM6214601
GSM6214576
GSM6214568
GSM6214594
GSM6214618
GSM6214567
GSM6214566
GSM6214593
GSM6214584
GSM6214616
GSM6214583
GSM6214617
GSM6214592
GSM6214585
GSM6214622
GSM6214572
GSM6214623
GSM6214624
GSM6214595
GSM6214599
GSM6214596
GSM6214598
GSM6214600
GSM6214564
GSM6214563
GSM6214582
GSM6214604
GSM6214605
GSM6214565
GSM6214580
GSM6214606
GSM6214571
GSM6214570
GSM6214581
GSM6214569
GSM6214586
GSM6214588
GSM6214587
PC2: 17% variance
PC1: 35% variance
10
0
-10
-20
-10
0
10
group
a DMSO
a adapalene
a tazarotene

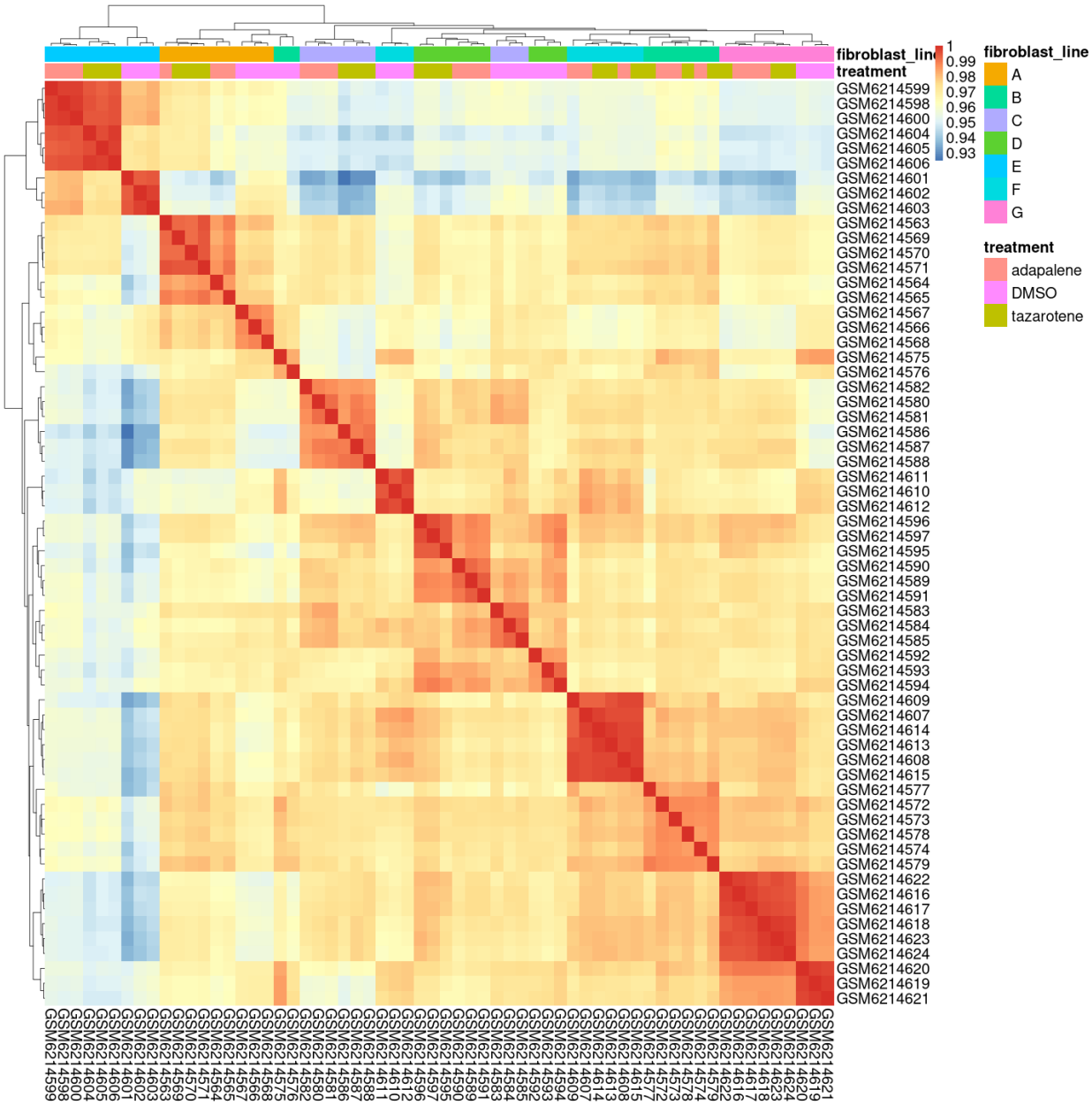
PCA - fibro line

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



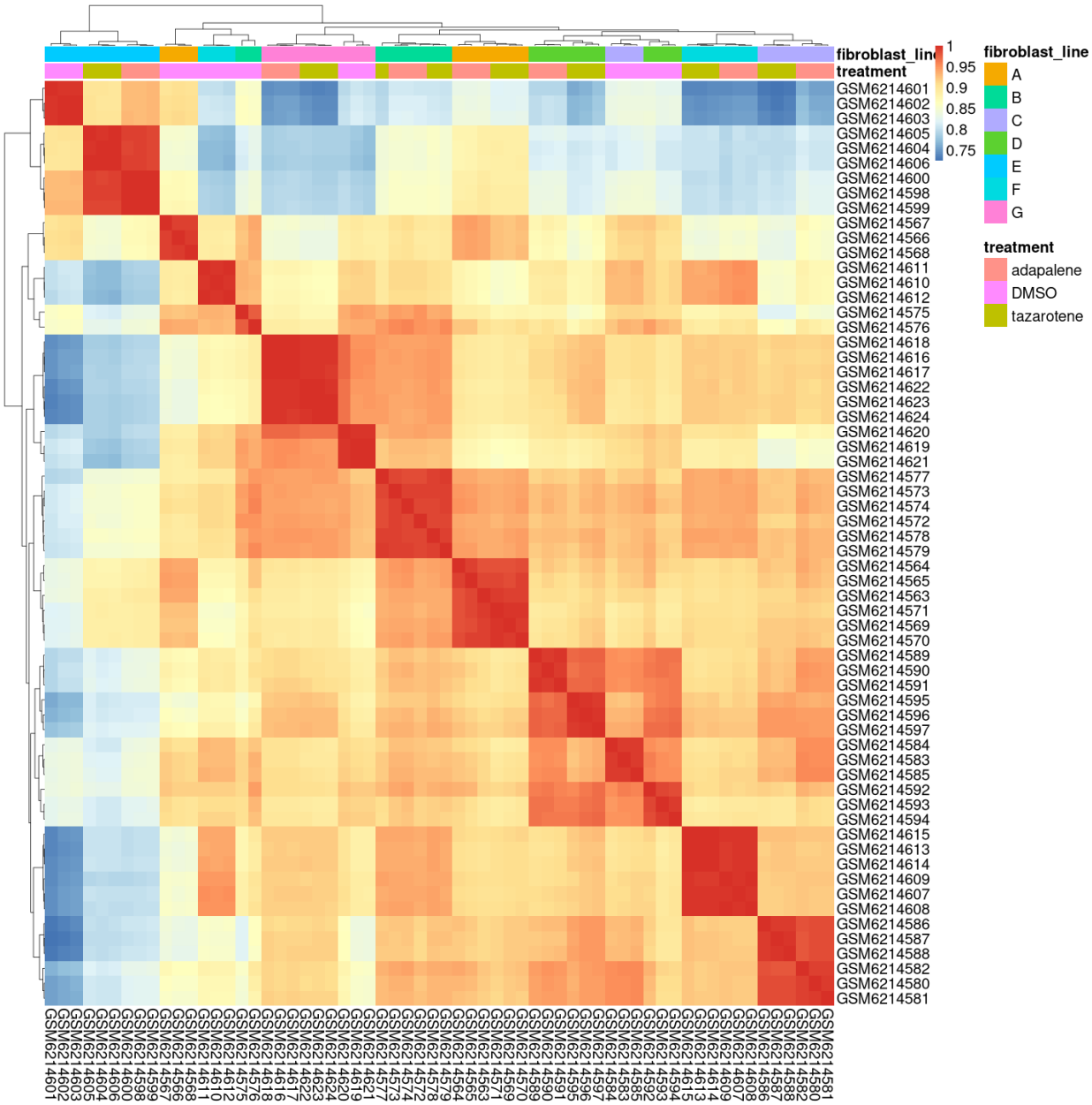
Inter-correlation analysis

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

# Create annotation file for samples
annotation <- metadata[, c("treatment", "fibroblast_line")]

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

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


fibroblast_line
treatment
1
0.99
0.98
0.97
0.96
0.95
0.94
0.93
fibroblast_line
A
B
C
D
E
F
G
treatment
adapalene
DMSO
tazarotene

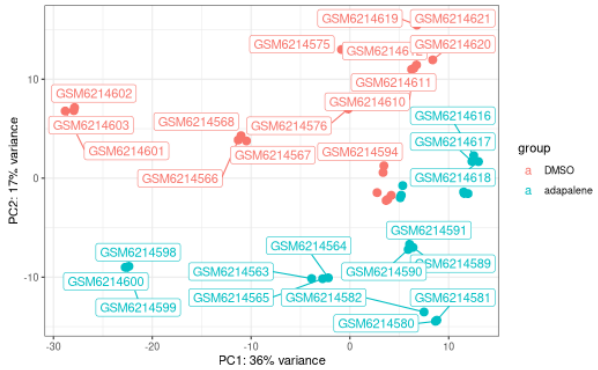
top 1000 variable genes

```
rv <- rowVars(rld_mat)
rv <- order(rv, decreasing = TRUE) %>% head(1000)
rld_mat_1000 <- rld_mat[rv,]
annotation <- metadata[, c("treatment", "fibroblast_line")]

# 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)
```


fibroblast_line
treatment
1
0.95
0.9
0.85
0.8
0.75
fibroblast_line
A
B
C
D
E
F
G
treatment
adapalene
DMSO
tazarotene

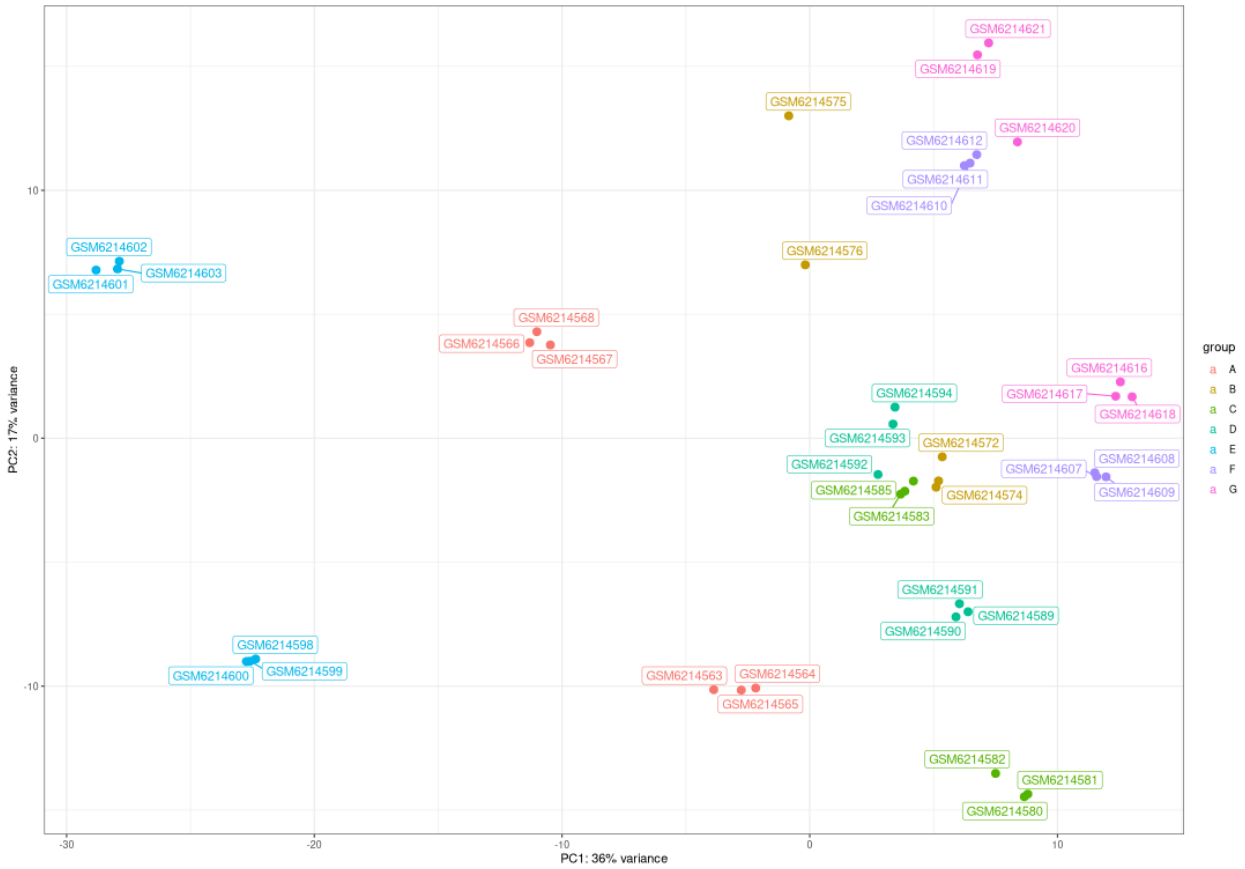
PCA: Treatment Adapalene vs DMSO

```
rld.sub <- rld[ , rld$treatment %in% c("adapalene", "DMSO") ]
plotPCA(rld.sub, intgroup = c("treatment")) + geom_label_repel(aes(label = name)) + theme_bw()
```



PCA: Treatment Adapalene vs DMSO

```
plotPCA(rld.sub, intgroup = c("fibroblast_line"))  + geom_label_repel(aes(label = name)) + theme_bw()
```




```
contrast <- c("treatment", "adapalene", "DMSO")
resTreatment <- results(dds, contrast = contrast, alpha = 0.05)
length(which(resTreatment$padj < 0.05))
```

```
## [1] 4232
```

```
# Add annotations
resTreatment_tb <- resTreatment %>%
  data.frame() %>%
  rownames_to_column(var = "gene") %>%
  as_tibble() %>%
```

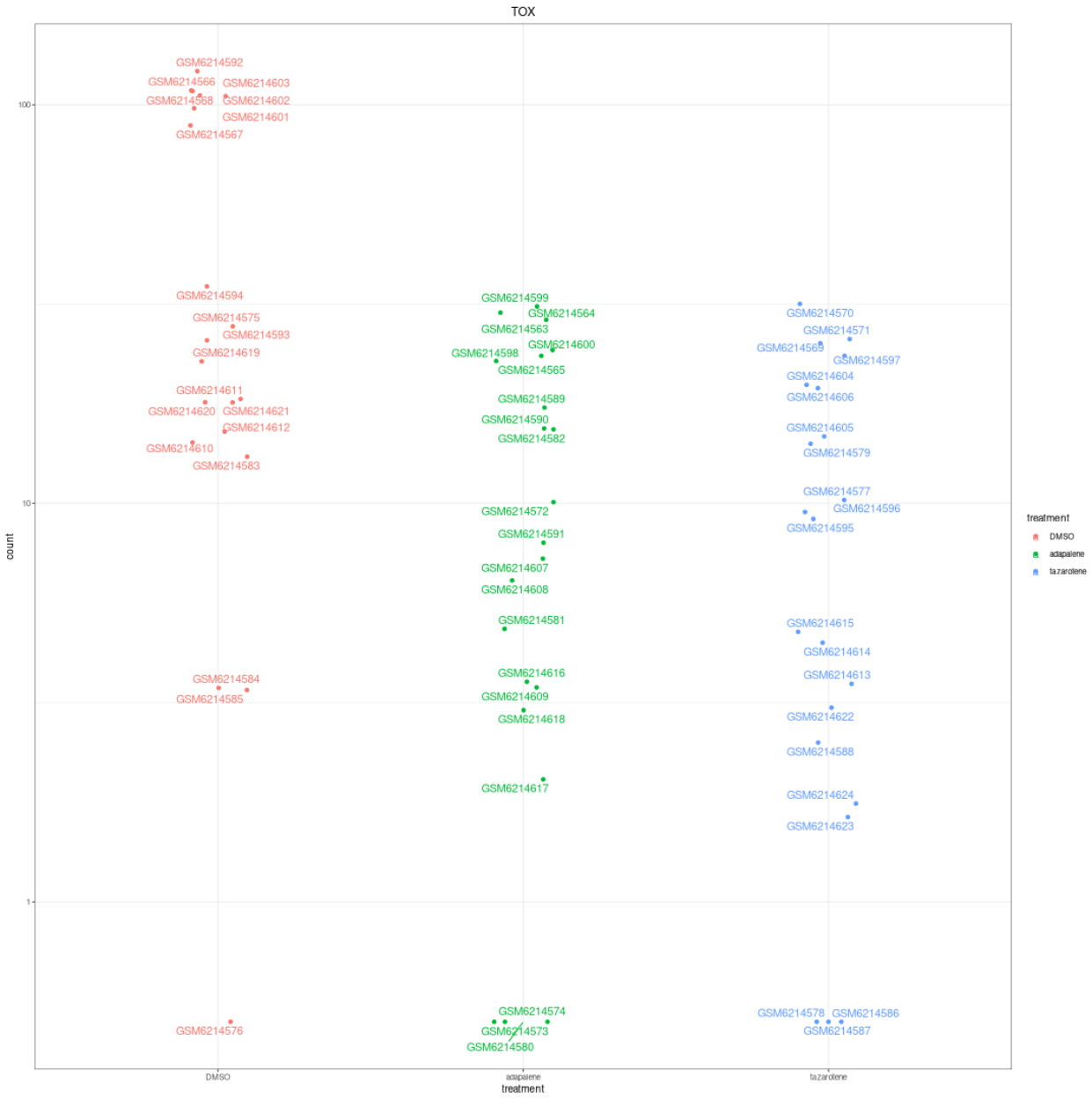
```
  left_join(gene_symbol, by = c("gene" = "gene_id"))

resTreatment_tb_significant <- dplyr::filter(resTreatment_tb, padj < 0.05) %>%
                        dplyr::filter(abs(log2FoldChange) > 1) %>%
                        comb_de_result_table()

samples_control <- metadata %>% rownames_to_column("ensembl_gene_id") %>%
         dplyr::filter(treatment == "DMSO") %>% pull("ensembl_gene_id")
```

```r
tpm_control <- get_counts_for_samples(counts_tpm, samples_control, "DMSO_mean_tpm")

samples_effect <- metadata %>% dplyr::filter(treatment == "adapalene") %>% row.names()
tpm_effect <- get_counts_for_samples(counts_tpm, samples_effect, "adapalene_tpm")

tpm_counts <- tpm_effect %>%
              left_join(tpm_control,
                        by = c("ensembl_gene_id" = "ensembl_gene_id"))

resTreatment_tb_significant <- resTreatment_tb_significant %>%
          left_join(tpm_counts, by = c("gene" = "ensembl_gene_id")) %>%
          arrange(log2FoldChange)

write_xlsx(list(T2.DE_adapalene = resTreatment_tb_significant),
          "tables/T2.DE_adapalene.xlsx")

# Separate into up and down-regulated gene sets
sigTreatment_up <- rownames(resTreatment)[which(resTreatment$padj < 0.01 & resTreatment$log2FoldChange >
sigTreatment_down <- rownames(resTreatment)[which(resTreatment$padj < 0.01 & resTreatment$log2FoldChang
```

Visualization

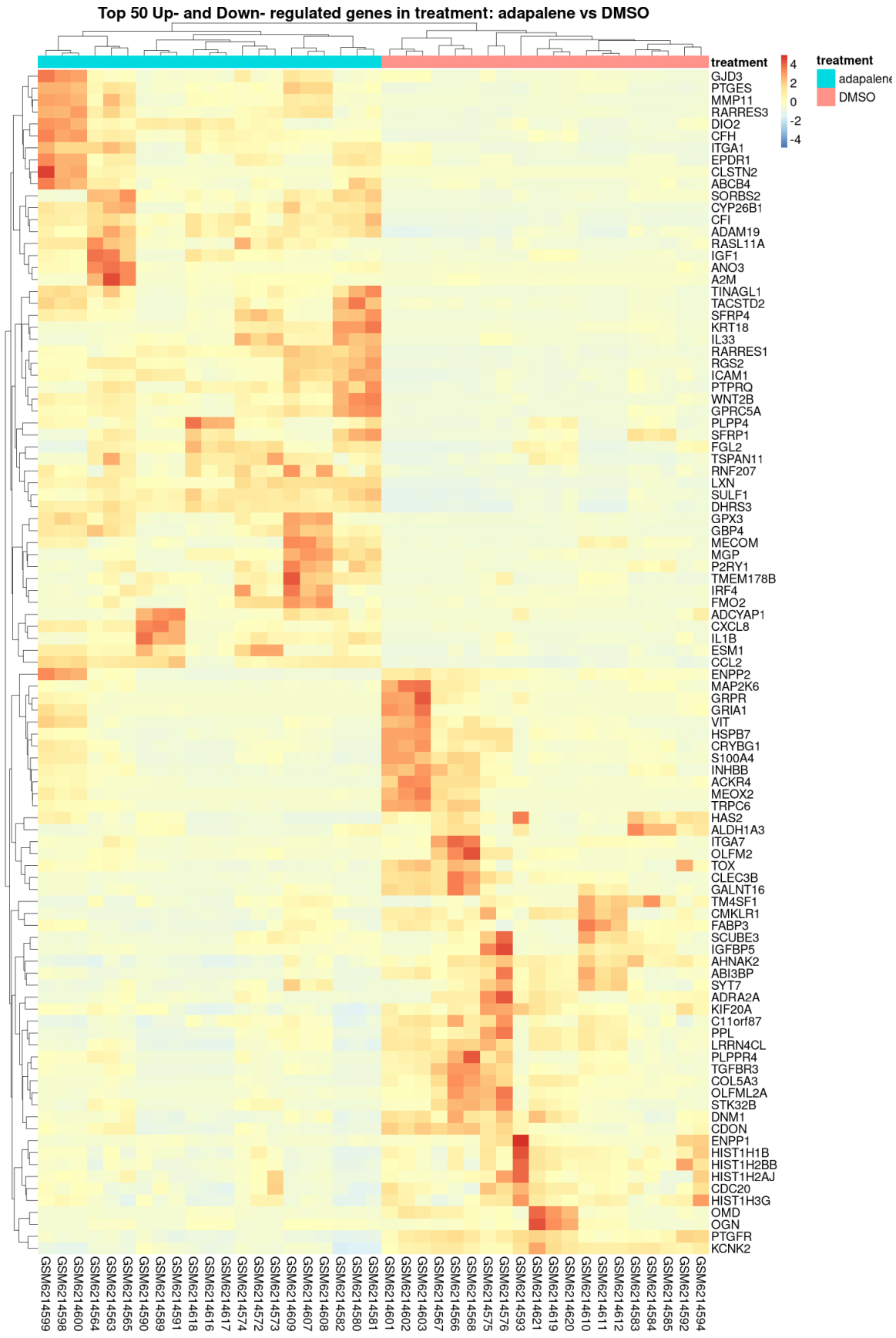
Gene example

```
d <- plotCounts(dds,
                gene = "ENSG00000198846",
                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("TOX") +
     theme(plot.title = element_text(hjust = 0.5)) +
     scale_y_log10()
```


TOX
GSM6214592
GSM6214566
GSM6214603
GSM6214588
GSM6214602
GSM6214601
GSM6214567
GSM6214594
GSM6214575
GSM6214593
GSM6214619
GSM6214611
GSM6214620
GSM6214621
GSM6214612
GSM6214610
GSM6214583
GSM6214584
GSM6214585
GSM6214576
GSM6214599
GSM6214564
GSM6214563
GSM6214600
GSM6214598
GSM6214565
GSM6214589
GSM6214590
GSM6214582
GSM6214572
GSM6214591
GSM6214607
GSM6214608
GSM6214581
GSM6214616
GSM6214609
GSM6214618
GSM6214617
GSM6214574
GSM6214573
GSM6214580
GSM6214570
GSM6214571
GSM6214569
GSM6214597
GSM6214604
GSM6214606
GSM6214605
GSM6214579
GSM6214577
GSM6214596
GSM6214595
GSM6214615
GSM6214614
GSM6214613
GSM6214622
GSM6214588
GSM6214624
GSM6214623
GSM6214578
GSM6214586
GSM6214587
count
100
10
1
DMSO
adapalene
tazarotene
treatment
treatment
DMSO
adapalene
tazarotene

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 <- counts_tpm %>% column_to_rownames("ensembl_gene_id") %>%
     dplyr::select(any_of(c(samples_control, samples_effect)))

plotmat <- plotmat[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 = metadata[, c("treatment"), drop = FALSE],
         main = "Top 50 Up- and Down- regulated genes in treatment: adapalene vs DMSO",
         fontsize = 20)
```


Top 50 Up- and Down- regulated genes in treatment: adapalene vs DMSO
treatment
adapalene
DMSO
4
2
0
-2
-4
GJD3
PTGES
MMP11
RARRES3
DIO2
CFH
ITGA1
EPDR1
CLSTN2
ABCB4
SORBS2
CYP26B1
CFI
ADAM19
RASL11A
IGF1
ANO3
A2M
TINAGL1
TACSTD2
SFRP4
KRT18
IL33
RARRES1
RGS2
ICAM1
PTPRQ
WNT2B
GPRC5A
PLPP4
SFRP1
FGL2
TSPAN11
RNF207
LXN
SULF1
DHRS3
GPX3
GBP4
MECOM
MGP
P2RY1
TMEM178B
IRF4
FMO2
ADCYAP1
CXCL8
IL1B
ESM1
CCL2
ENPP2
MAP2K6
GRPR
GRIA1
VIT
HSPB7
CRYBG1
S100A4
INHBB
ACKR4
MEOX2
TRPC6
HAS2
ALDH1A3
ITGA7
OLFM2
TOX
CLEC3B
GALNT16
TM4SF1
CMKLR1
FABP3
SCUBE3
IGFBP5
AHNAK2
ABI3BP
SYT7
ADRA2A
KIF20A
C11orf87
PPL
LRRN4CL
PLPPR4
TGFBR3
COL5A3
OLFML2A
STK32B
DNM1
CDON
ENPP1
HIST1H1B
HIST1H2BB
HIST1H2AJ
CDC20
HIST1H3G
OMD
OGN
PTGFR
KCNK2
GSM6214599
GSM6214598
GSM6214600
GSM6214564
GSM6214563
GSM6214565
GSM6214590
GSM6214589
GSM6214591
GSM6214618
GSM6214616
GSM6214617
GSM6214574
GSM6214572
GSM6214573
GSM6214609
GSM6214607
GSM6214608
GSM6214582
GSM6214580
GSM6214581
GSM6214601
GSM6214602
GSM6214603
GSM6214567
GSM6214566
GSM6214568
GSM6214575
GSM6214576
GSM6214593
GSM6214621
GSM6214619
GSM6214620
GSM6214610
GSM6214611
GSM6214612
GSM6214583
GSM6214584
GSM6214585
GSM6214592
GSM6214594

R session

```
sessionInfo()
```

```
## R version 4.2.2 (2022-10-31)
## Platform: x86_64-redhat-linux-gnu (64-bit)
## Running under: Fedora Linux 37 (Workstation Edition)
##
## Matrix products: default
## BLAS/LAPACK: /usr/lib64/libflexiblas.so.3.3
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] writexl_1.4.1               ggplotify_0.1.0
##  [3] knitr_1.41                  ggrepel_0.9.2
##  [5] tximport_1.26.1             DEGreport_1.34.0
##  [7] pheatmap_1.0.12             DESeq2_1.38.2
##  [9] SummarizedExperiment_1.28.0 MatrixGenerics_1.10.0
## [11] matrixStats_0.63.0          RColorBrewer_1.1-3
## [13] ensembldb_2.22.0            AnnotationFilter_1.22.0
## [15] GenomicFeatures_1.50.3      AnnotationDbi_1.60.0
## [17] Biobase_2.58.0              GenomicRanges_1.50.2
## [19] GenomeInfoDb_1.34.9         IRanges_2.32.0
## [21] S4Vectors_0.36.1            AnnotationHub_3.6.0
## [23] BiocFileCache_2.6.0         dbplyr_2.2.1
## [25] BiocGenerics_0.44.0         forcats_0.5.2
## [27] stringr_1.5.0               dplyr_1.0.10
## [29] purrr_1.0.0                 readr_2.1.3
## [31] tidyr_1.2.1                 tibble_3.1.8
## [33] ggplot2_3.4.0               tidyverse_1.3.2
##
## loaded via a namespace (and not attached):
##   [1] utf8_1.2.2                   tidyselect_1.2.0
##   [3] RSQLite_2.2.20               grid_4.2.2
##   [5] BiocParallel_1.32.5          munsell_0.5.0
##   [7] codetools_0.2-18             withr_2.5.0
##   [9] colorspace_2.0-3             filelock_1.0.2
##  [11] highr_0.10                   rstudioapi_0.14
##  [13] labeling_0.4.2               GenomeInfoDbData_1.2.9
##  [15] mnormt_2.1.1                 bit64_4.0.5
##  [17] farver_2.1.1                 vctrs_0.5.1
##  [19] generics_0.1.3               xfun_0.36
##  [21] timechange_0.1.1             R6_2.5.1
```

```
##  [23] doParallel_1.0.17           clue_0.3-63
##  [25] locfit_1.5-9.7              bitops_1.0-7
##  [27] cachem_1.0.6                reshape_0.8.9
##  [29] gridGraphics_0.5-1          DelayedArray_0.24.0
##  [31] assertthat_0.2.1            promises_1.2.0.1
##  [33] BiocIO_1.8.0                scales_1.2.1
##  [35] vroom_1.6.0                 googlesheets4_1.0.1
##  [37] gtable_0.3.1                rlang_1.0.6
##  [39] MatrixModels_0.5-2          GlobalOptions_0.1.2
##  [41] splines_4.2.2               rtracklayer_1.58.0
##  [43] lazyeval_0.2.2              gargle_1.2.0
##  [45] broom_1.0.2                 BiocManager_1.30.19
##  [47] yaml_2.3.6                  modelr_0.1.10
##  [49] backports_1.4.1             httpuv_1.6.7
##  [51] tools_4.2.2                 psych_2.2.9
##  [53] logging_0.10-108            ellipsis_0.3.2
##  [55] ggdendro_0.1.23             Rcpp_1.0.9
##  [57] plyr_1.8.8                  progress_1.2.2
##  [59] zlibbioc_1.44.0             RCurl_1.98-1.8
##  [61] prettyunits_1.1.1           GetoptLong_1.0.5
##  [63] cowplot_1.1.1               haven_2.5.1
##  [65] cluster_2.1.4               fs_1.5.2
##  [67] magrittr_2.0.3              SparseM_1.81
##  [69] circlize_0.4.15             reprex_2.0.2
##  [71] googledrive_2.0.0           ProtGenerics_1.30.0
##  [73] hms_1.1.2                   mime_0.12
##  [75] evaluate_0.19               xtable_1.8-4
##  [77] XML_3.99-0.13               readxl_1.4.1
##  [79] shape_1.4.6                 compiler_4.2.2
##  [81] biomaRt_2.54.0              crayon_1.5.2
##  [83] htmltools_0.5.4             later_1.3.0
##  [85] tzdb_0.3.0                  geneplotter_1.76.0
##  [87] lubridate_1.9.0             DBI_1.1.3
##  [89] ComplexHeatmap_2.14.0       MASS_7.3-58.1
##  [91] rappdirs_0.3.3              Matrix_1.6-1
##  [93] cli_3.5.0                   parallel_4.2.2
##  [95] pkgconfig_2.0.3             GenomicAlignments_1.34.0
##  [97] xml2_1.3.3                  foreach_1.5.2
##  [99] annotate_1.76.0             XVector_0.38.0
## [101] rvest_1.0.3                 yulab.utils_0.0.6
## [103] digest_0.6.31               ConsensusClusterPlus_1.62.0
## [105] Biostrings_2.66.0           rmarkdown_2.19
## [107] cellranger_1.1.0            edgeR_3.40.1
## [109] restfulr_0.0.15             curl_4.3.2
## [111] shiny_1.7.4                 Rsamtools_2.14.0
## [113] quantreg_5.97               rjson_0.2.21
## [115] lifecycle_1.0.3             nlme_3.1-160
## [117] jsonlite_1.8.4              limma_3.54.0
## [119] fansi_1.0.3                 pillar_1.8.1
## [121] lattice_0.20-45             KEGGREST_1.38.0
## [123] fastmap_1.1.0               httr_1.4.4
## [125] survival_3.4-0              interactiveDisplayBase_1.36.0
## [127] glue_1.6.2                  png_0.1-8
## [129] iterators_1.0.14            BiocVersion_3.16.0
```

```
## [131] bit_4.0.5                    stringi_1.7.8
## [133] blob_1.2.3                   memoise_2.0.1
```