RNA-seq course- week1

Serhiy Naumenko

2024-07-27

Contents

Overview

- Petrenko2024 RNA-seq experiment for reanalysis

Load Counts

```
# raw counts downloaded from
# https://www.ebi.ac.uk/biostudies/arrayexpress/studies/E-MTAB-13804

# setwd("02_classes/03_rnaseq_intro_part1/")
counts_csv <- "../../01_data/count_matrix_raw.csv"
counts_tpm_csv <- "../../01_data/count_matrix_tpm.csv"

if (file.exists(counts_csv)){
```

```
    counts <- read_csv(counts_csv)
    colnames(counts)[1] <- "gene_name"
    counts_tpm <- read_csv(counts_tpm_csv)
}
```

```
## Error: '../../01_data/count_matrix_tpm.csv' does not exist in current working directory ('/home/serhi
```

Load metadata

```
# Load the data and metadata
# remove duplicate rows
metadata <- read_tsv("../../01_data/E-MTAB-13804.sdrf.txt") %>%
    dplyr::select(-any_of(c("Scan Name", "Comment[SUBMITTED_FILE_NAME]",
                            "Comment[ENA_RUN]", "Comment[FASTQ_URI]"))) %>% distinct() %>%
    column_to_rownames(var = "Source Name")
protein_coding_genes <- read_csv("../../99_technical/ensembl_w_description.mouse.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.

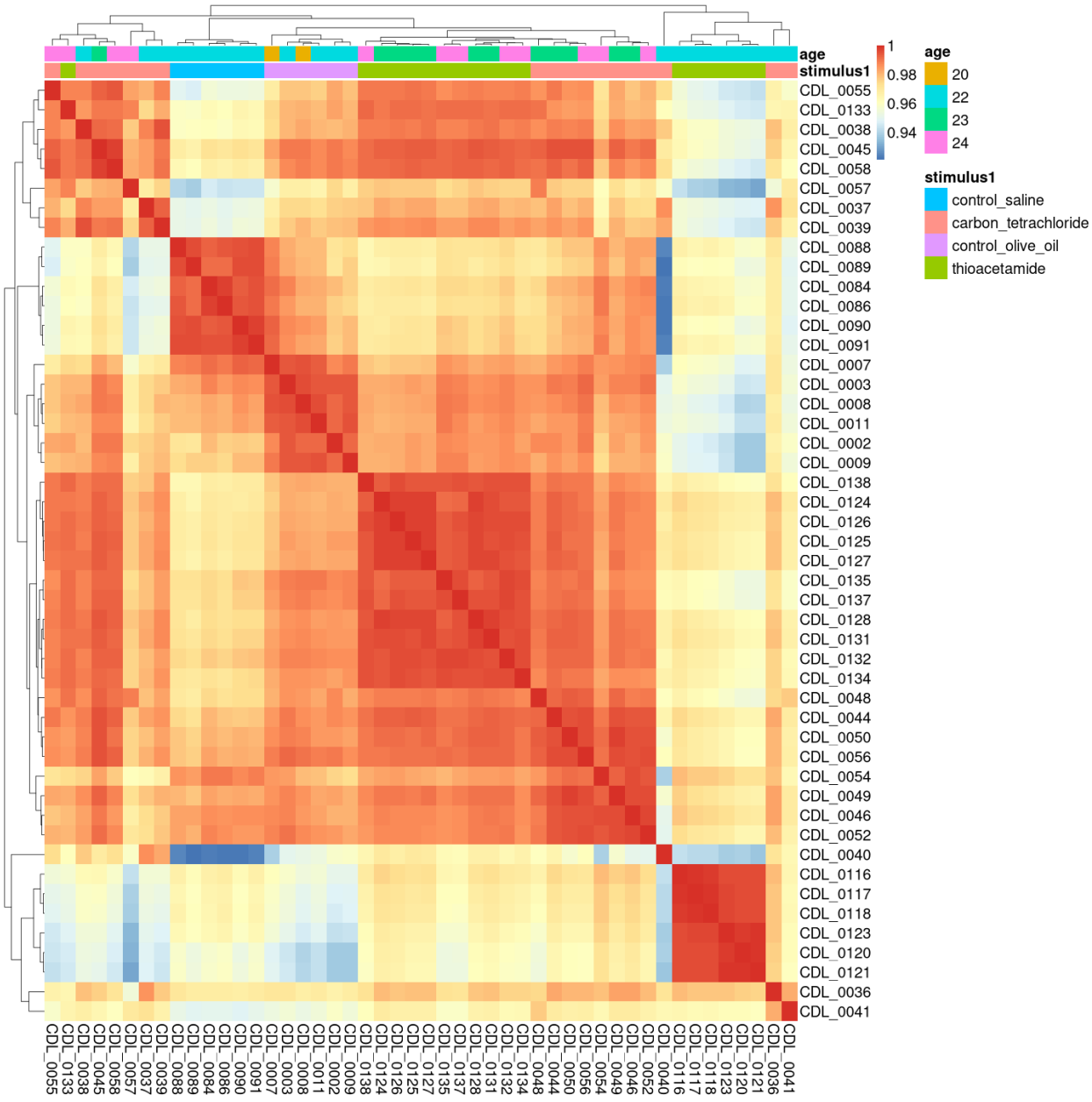
Sample-level QC analysis

PCA - stimulus1

```
# Use the DESeq2 function
plotPCA(rld, intgroup = c("stimulus1")) + geom_label_repel(aes(label = name)) + theme_bw()
```

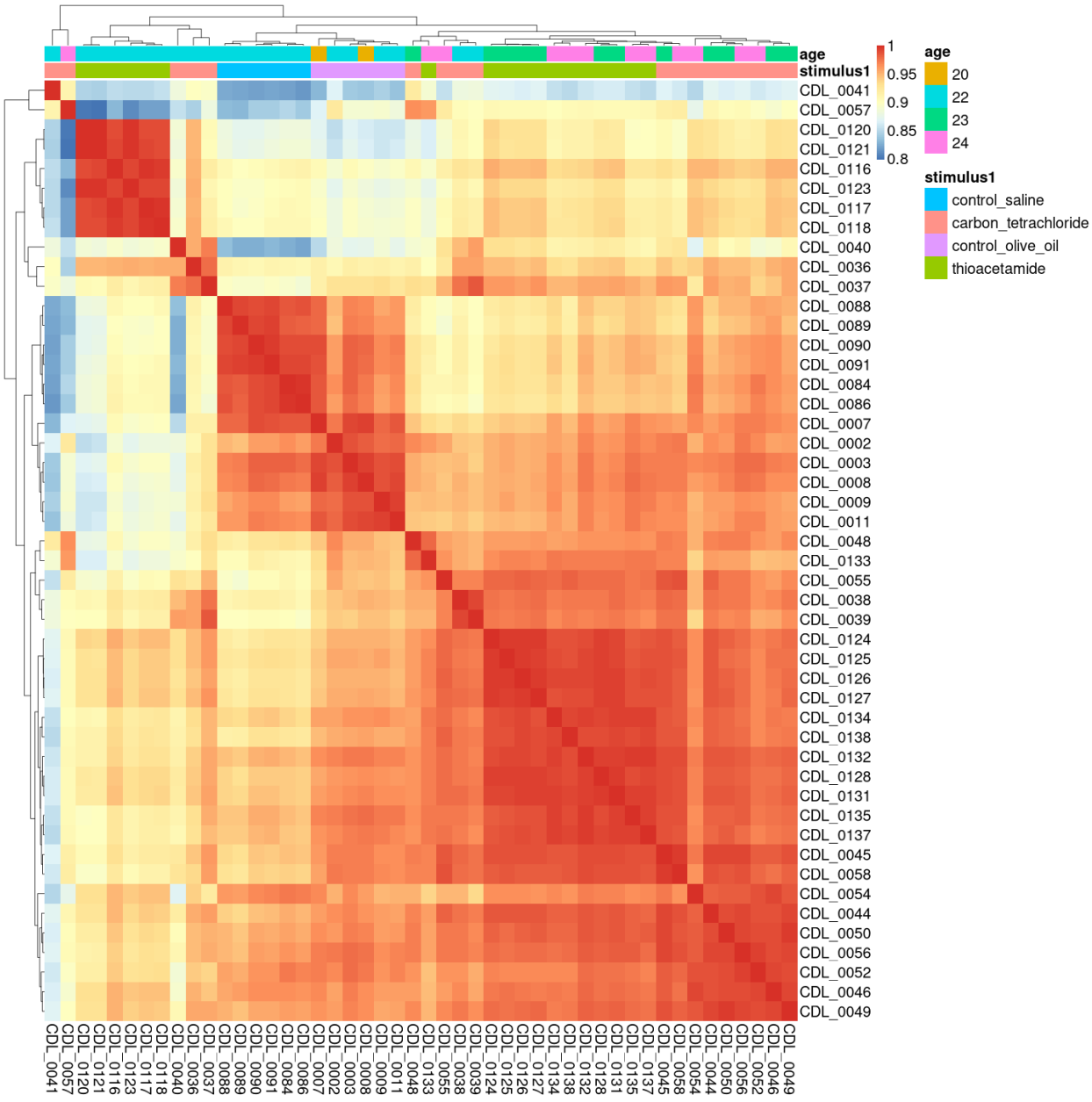
Inter-correlation analysis

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

# Create annotation file for samples
annotation <- metadata_prep[, c("stimulus1", "age")]

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

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


age
stimulus1
CDL_0055
CDL_0133
CDL_0038
CDL_0045
CDL_0058
CDL_0057
CDL_0037
CDL_0039
CDL_0088
CDL_0089
CDL_0084
CDL_0086
CDL_0090
CDL_0091
CDL_0007
CDL_0003
CDL_0008
CDL_0011
CDL_0002
CDL_0009
CDL_0138
CDL_0124
CDL_0126
CDL_0125
CDL_0127
CDL_0135
CDL_0137
CDL_0128
CDL_0131
CDL_0132
CDL_0134
CDL_0048
CDL_0044
CDL_0050
CDL_0056
CDL_0054
CDL_0049
CDL_0046
CDL_0052
CDL_0040
CDL_0116
CDL_0117
CDL_0118
CDL_0123
CDL_0120
CDL_0121
CDL_0036
CDL_0041
1
0.98
0.96
0.94
age
20
22
23
24
stimulus1
control_saline
carbon_tetrachloride
control_olive_oil
thioacetamide

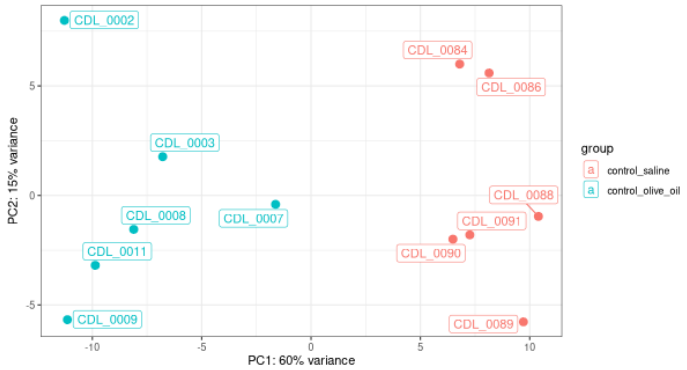
top 1000 variable genes

```
rv <- rowVars(rld_mat)
rv <- order(rv, decreasing = TRUE) %>% head(1000)
rld_mat_1000 <- rld_mat[rv,]
annotation <- metadata_prep[, c("stimulus1", "age")]

# 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)
```


age
stimulus1
CDL_0041
CDL_0057
CDL_0120
CDL_0121
CDL_0116
CDL_0123
CDL_0117
CDL_0118
CDL_0040
CDL_0036
CDL_0037
CDL_0088
CDL_0089
CDL_0090
CDL_0091
CDL_0084
CDL_0086
CDL_0007
CDL_0002
CDL_0003
CDL_0008
CDL_0009
CDL_0011
CDL_0048
CDL_0133
CDL_0055
CDL_0038
CDL_0039
CDL_0124
CDL_0125
CDL_0126
CDL_0127
CDL_0134
CDL_0138
CDL_0132
CDL_0128
CDL_0131
CDL_0135
CDL_0137
CDL_0045
CDL_0058
CDL_0054
CDL_0044
CDL_0050
CDL_0056
CDL_0052
CDL_0046
CDL_0049
1
0.95
0.9
0.85
0.8
age
20
22
23
24
stimulus1
control_saline
carbon_tetrachloride
control_olive_oil
thioacetamide

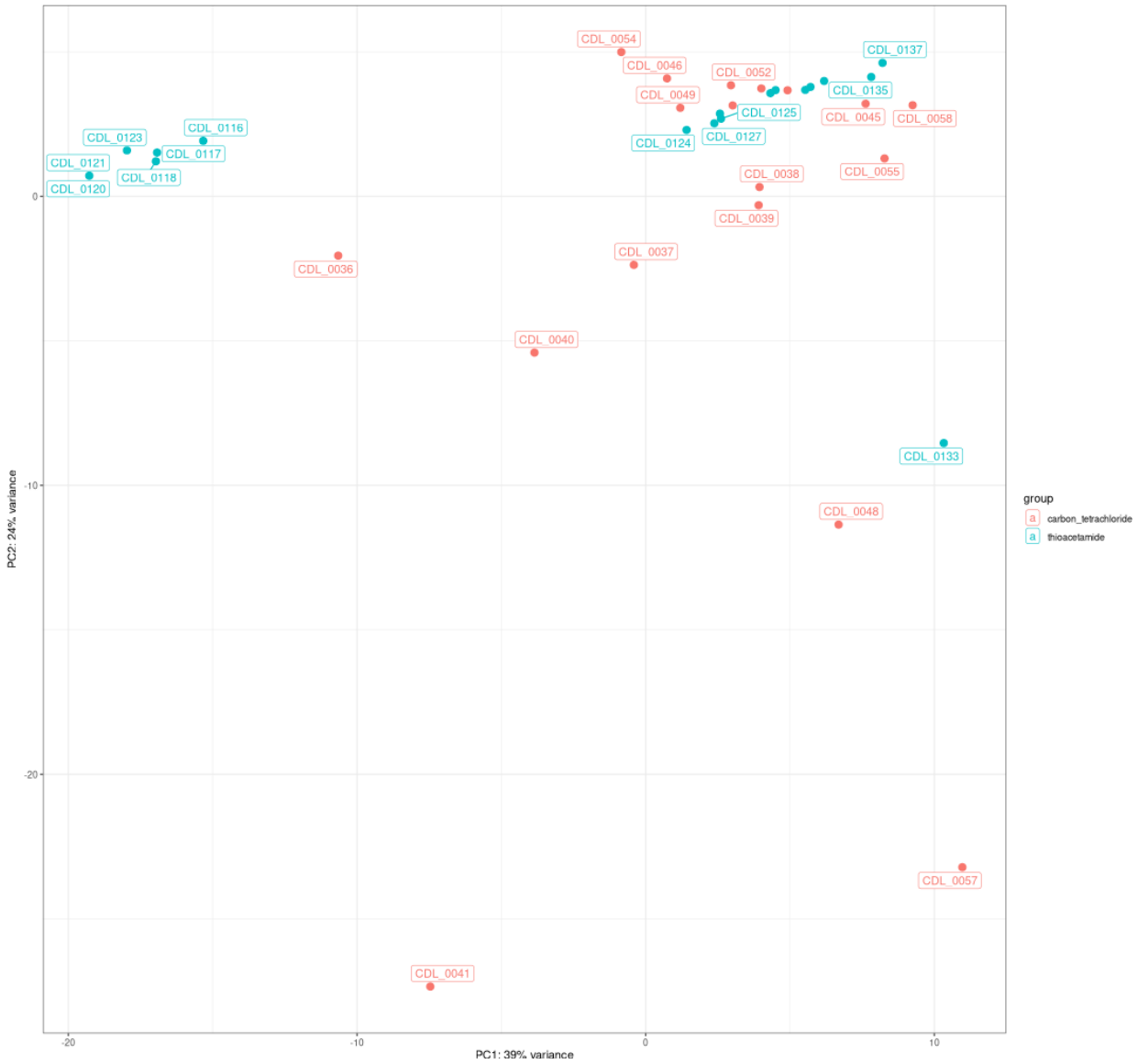
PCA: Controls

```
rld.sub <- rld[ , rld$stimulus1 %in% c("control_saline", "control_olive_oil") ]
plotPCA(rld.sub, intgroup = c("stimulus1")) + geom_label_repel(aes(label = name)) + theme_bw()
```



PCA: treatment

```
rld.sub <- rld[ , rld$stimulus1 %in% c("carbon_tetrachloride", "thioacetamide") ]
plotPCA(rld.sub, intgroup = c("stimulus1")) + 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))

# 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")
```

```
#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$padj < 0.01 & resTreatment$log2FoldChan
```

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()
```

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)
```

R session

```
sessionInfo()
```

```
## R version 4.4.1 (2024-06-14)
## Platform: x86_64-redhat-linux-gnu
## Running under: Fedora Linux 40 (Workstation Edition)
##
## Matrix products: default
## BLAS/LAPACK: FlexiBLAS OPENBLAS-OPENMP;  LAPACK version 3.11.0
##
## 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
##
## time zone: America/Toronto
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] writexl_1.5.0               ggplotify_0.1.2
##  [3] knitr_1.48                  ggrepel_0.9.5
##  [5] tximport_1.32.0             DEGreport_1.40.1
##  [7] pheatmap_1.0.12             DESeq2_1.44.0
##  [9] SummarizedExperiment_1.34.0 MatrixGenerics_1.16.0
## [11] matrixStats_1.3.0           RColorBrewer_1.1-3
## [13] ensembldb_2.28.0            AnnotationFilter_1.28.0
## [15] GenomicFeatures_1.56.0      AnnotationDbi_1.66.0
## [17] Biobase_2.64.0              GenomicRanges_1.56.1
## [19] GenomeInfoDb_1.40.1         IRanges_2.38.1
## [21] S4Vectors_0.42.1            AnnotationHub_3.12.0
## [23] BiocFileCache_2.12.0        dbplyr_2.5.0
## [25] BiocGenerics_0.50.0         lubridate_1.9.3
## [27] forcats_1.0.0               stringr_1.5.1
## [29] dplyr_1.1.4                 purrr_1.0.2
## [31] readr_2.1.5                 tidyr_1.3.1
## [33] tibble_3.2.1                ggplot2_3.5.1
## [35] tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##   [1] ggdendro_0.2.0              rstudioapi_0.16.0
##   [3] jsonlite_1.8.8              shape_1.4.6.1
##   [5] magrittr_2.0.3              farver_2.1.2
##   [7] rmarkdown_2.27              fs_1.6.4
##   [9] GlobalOptions_0.1.2         BiocIO_1.14.0
##  [11] zlibbioc_1.50.0             vctrs_0.6.5
##  [13] memoise_2.0.1               Rsamtools_2.20.0
##  [15] RCurl_1.98-1.16             htmltools_0.5.8.1
##  [17] S4Arrays_1.4.1              curl_5.2.1
##  [19] broom_1.0.6                 gridGraphics_0.5-1
##  [21] SparseArray_1.4.8           plyr_1.8.9
##  [23] cachem_1.1.0                GenomicAlignments_1.40.0
##  [25] lifecycle_1.0.4             iterators_1.0.14
##  [27] pkgconfig_2.0.3             Matrix_1.7-0
##  [29] R6_2.5.1                    fastmap_1.2.0
##  [31] GenomeInfoDbData_1.2.12     clue_0.3-65
##  [33] digest_0.6.36               reshape_0.8.9
##  [35] colorspace_2.1-0            RSQLite_2.3.7
##  [37] labeling_0.4.3              filelock_1.0.3
##  [39] fansi_1.0.6                 timechange_0.3.0
##  [41] httr_1.4.7                  abind_1.4-5
##  [43] compiler_4.4.1              bit64_4.0.5
##  [45] withr_3.0.0                 doParallel_1.0.17
##  [47] ConsensusClusterPlus_1.68.0 backports_1.5.0
```

```
##  [49] BiocParallel_1.38.0         DBI_1.2.3
##  [51] psych_2.4.6.26              highr_0.11
##  [53] MASS_7.3-60.2               rappdirs_0.3.3
##  [55] DelayedArray_0.30.1         rjson_0.2.21
##  [57] tools_4.4.1                 glue_1.7.0
##  [59] restfulr_0.0.15             nlme_3.1-164
##  [61] grid_4.4.1                  cluster_2.1.6
##  [63] generics_0.1.3              gtable_0.3.5
##  [65] tzdb_0.4.0                  hms_1.1.3
##  [67] utf8_1.2.4                  XVector_0.44.0
##  [69] BiocVersion_3.19.1          foreach_1.5.2
##  [71] pillar_1.9.0                vroom_1.6.5
##  [73] yulab.utils_0.1.5           limma_3.60.4
##  [75] logging_0.10-108            circlize_0.4.16
##  [77] lattice_0.22-6              rtracklayer_1.64.0
##  [79] bit_4.0.5                   tidyselect_1.2.1
##  [81] ComplexHeatmap_2.20.0       locfit_1.5-9.10
##  [83] Biostrings_2.72.1           ProtGenerics_1.36.0
##  [85] edgeR_4.2.1                 xfun_0.45
##  [87] statmod_1.5.0               stringi_1.8.4
##  [89] UCSC.utils_1.0.0            lazyeval_0.2.2
##  [91] yaml_2.3.9                  evaluate_0.24.0
##  [93] codetools_0.2-20            BiocManager_1.30.23
##  [95] cli_3.6.3                   munsell_0.5.1
##  [97] Rcpp_1.0.13                 png_0.1-8
##  [99] XML_3.99-0.17               parallel_4.4.1
## [101] blob_1.2.4                  bitops_1.0-7
## [103] scales_1.3.0                crayon_1.5.3
## [105] GetoptLong_1.0.5            rlang_1.1.4
## [107] mnormt_2.1.1                cowplot_1.1.3
## [109] KEGGREST_1.44.1
```