Differential expression analysis of Cai's samples for TK_27

Hernan Lorenzi

8/11/2022

Load libraries

```
suppressMessages(library("org.Hs.eg.db"))
```

```
## Warning: package 'AnnotationDbi' was built under R version 4.1.2
```

```
## Warning: package 'S4Vectors' was built under R version 4.1.3
```

```
suppressMessages(library("pheatmap"))
suppressMessages(library("EnhancedVolcano"))
suppressMessages(library("ggplot2"))
suppressMessages(library("ggpubr"))
suppressMessages(library("DESeq2"))
```

```
## Warning: package 'GenomicRanges' was built under R version 4.1.2
```

```
## Warning: package 'GenomeInfoDb' was built under R version 4.1.2
```

```
## Warning: package 'matrixStats' was built under R version 4.1.2
```

```
suppressMessages(library("stringr"))
suppressMessages(library("biomaRt"))
```

```
## Warning: package 'biomaRt' was built under R version 4.1.2
```

```
suppressMessages(library("tidyverse"))
```

```
## Warning: package 'tibble' was built under R version 4.1.2
```

```
## Warning: package 'tidyr' was built under R version 4.1.2
```

```
## Warning: package 'readr' was built under R version 4.1.2
```

```
## Warning: package 'dplyr' was built under R version 4.1.2
```

```
suppressMessages(library("pcaExplorer"))
```

```
## Warning: package 'pcaExplorer' was built under R version 4.1.3
```

```
suppressMessages(library("VennDiagram"))
suppressMessages(library("clusterProfiler"))
```

```
## Warning: package 'clusterProfiler' was built under R version 4.1.2
```

```
suppressMessages(library("GOSemSim"))
suppressMessages(library("ggsci"))
suppressMessages(library("viridis"))
suppressMessages(library("ggrepel"))
suppressMessages(library("RColorBrewer"))
```

```
## Warning: package 'RColorBrewer' was built under R version 4.1.2
```

```
suppressMessages(library("msigdbr"))
```

```
## Warning: package 'msigdbr' was built under R version 4.1.2
```

```
suppressMessages(library("cowplot"))
suppressMessages(library("enrichplot"))
```

```
## Warning: package 'enrichplot' was built under R version 4.1.2
```

```
suppressMessages(library("ReactomePA"))
suppressMessages(library("ggupset"))
suppressMessages(library("broom"))
suppressMessages(library("ggraph"))
```

Define functions

```
# Load auxyliary functions
source(file = "./01_aux_rnaseq_functions.R")

# Load enrichment functions
source(file = "./02_Gene_enrichment_functions.R")
```

Load data

```
all <- read.delim2("./data/read_counts.txt", sep = "\t", header = TRUE, row.names = 1, comment.char = c

# Make sure read counts are numeric and rounded to 0 decimals
all.tmp <- as.data.frame(lapply(all, function(x){ round(as.numeric(x), digits = 0)} ))
rownames(all.tmp) <- rownames(all)
all <- all.tmp

# Replace NA counts with 0
all[is.na(all)] <- 0

# Keep table with Ensemble IDs and gene Symbols
gene_symbols <- replace_gene_acc_by_symbol_ids(rownames(all))
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
ensembl_to_symbol <- as.data.frame(cbind("Ensembl_ID" = rownames(all), "gene_name" = gene_symbols), row

# Load metadata
metadata <- read.delim2("./data/metadata.txt", sep = "\t", row.names = 1, header = T)

# Sort tables so metadata and read counts match order
metadata<-  metadata[match(colnames(all), rownames(metadata)), ]

# Add total read counts and sample id columns to metadata
metadata$Read_counts <- colSums(all)

#Remove all zero rows
all <- remove_all_zero_rows(all, min_total_count = 0)
```

Normalize data to TPMs to run some comparative analysis across samples

```
all.tpm <- normalize_by_TPM(all)
write.table(x = all.tpm, file = "./data/read_counts_tpms.txt", col.names = NA, sep = "\t")
```

Analysis of expression data using DESeq2

```
# Convert metadata to factors
for (variable in c("genotype", "sample_id")){
  metadata[,variable] <- as.factor(metadata[,variable])
}
```

Analysis of Dataset ONE

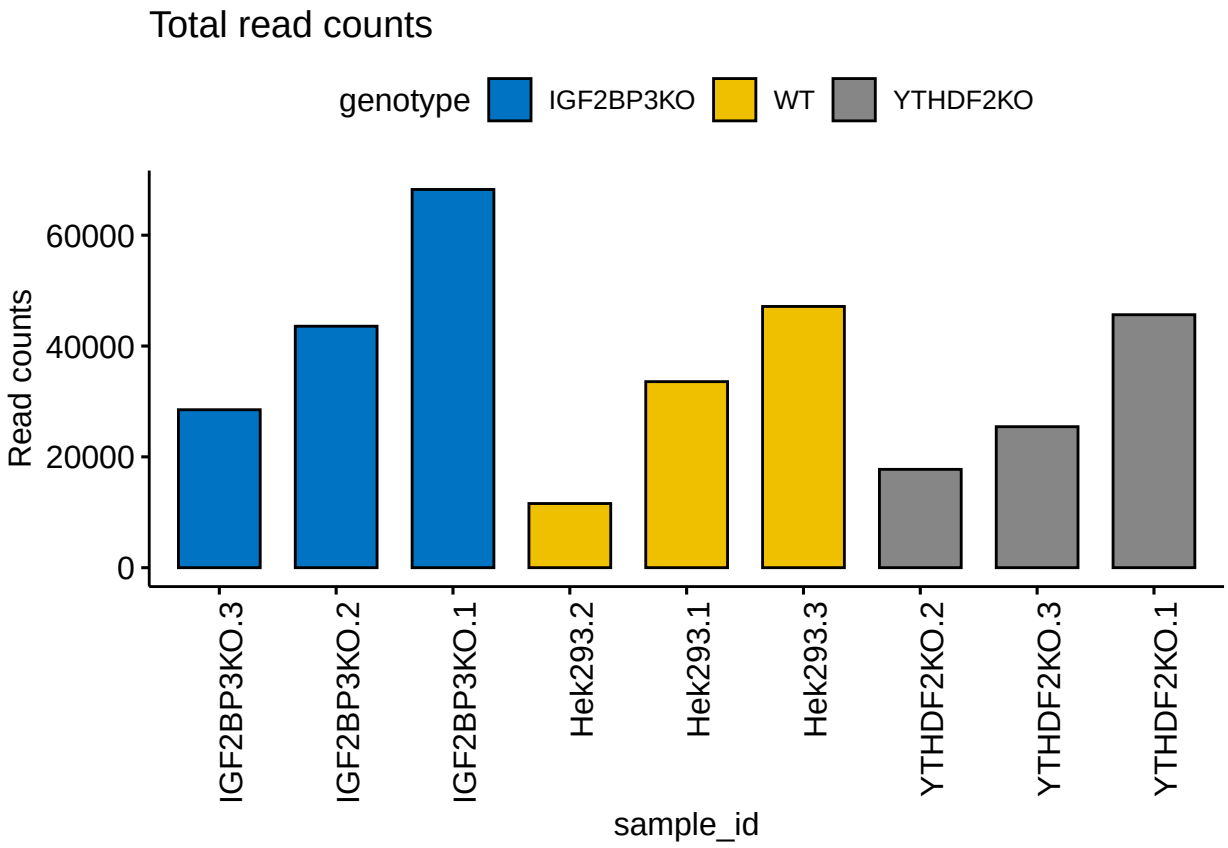
```r
# Generate DESeq2 object for NS and ST condition ONLY. We could potentially add Read_counts as either a
#dds.all <- DESeqDataSetFromMatrix(countData = all_one,
#                                  colData = meta_one,
#                                  design = ~ Genotype + Inducer + Genotype:Inducer)

dir.create(path = "./Plots", showWarnings = FALSE)


# Create DESeq object
dds.all <- DESeqDataSetFromMatrix(countData = all,
                                  colData = metadata,
                                  design = ~ Read_counts + genotype)
```

```
## converting counts to integer mode

##   the design formula contains one or more numeric variables with integer values,
##   specifying a model with increasing fold change for higher values.
##   did you mean for this to be a factor? if so, first convert
##   this variable to a factor using the factor() function

##   the design formula contains one or more numeric variables that have mean or
##   standard deviation larger than 5 (an arbitrary threshold to trigger this message).
##   Including numeric variables with large mean can induce collinearity with the intercept.
##   Users should center and scale numeric variables in the design to improve GLM convergence.
```

```r
# Plot total reads per sample using barchar
p <- ggbarplot(data = metadata,
          x = "sample_id",
          y = "Read_counts",
          x.text.angle = 90,
          fill = "genotype",
          title = "Total read counts",
          ylab = "Read counts",
          sort.by.groups = TRUE,
          palette = "jco",
          sort.val = "asc")
ggsave("Plots/barplot_read_counts.pdf", plot = p)
```

```
## Saving 6.5 x 4.5 in image
```

```
p
```

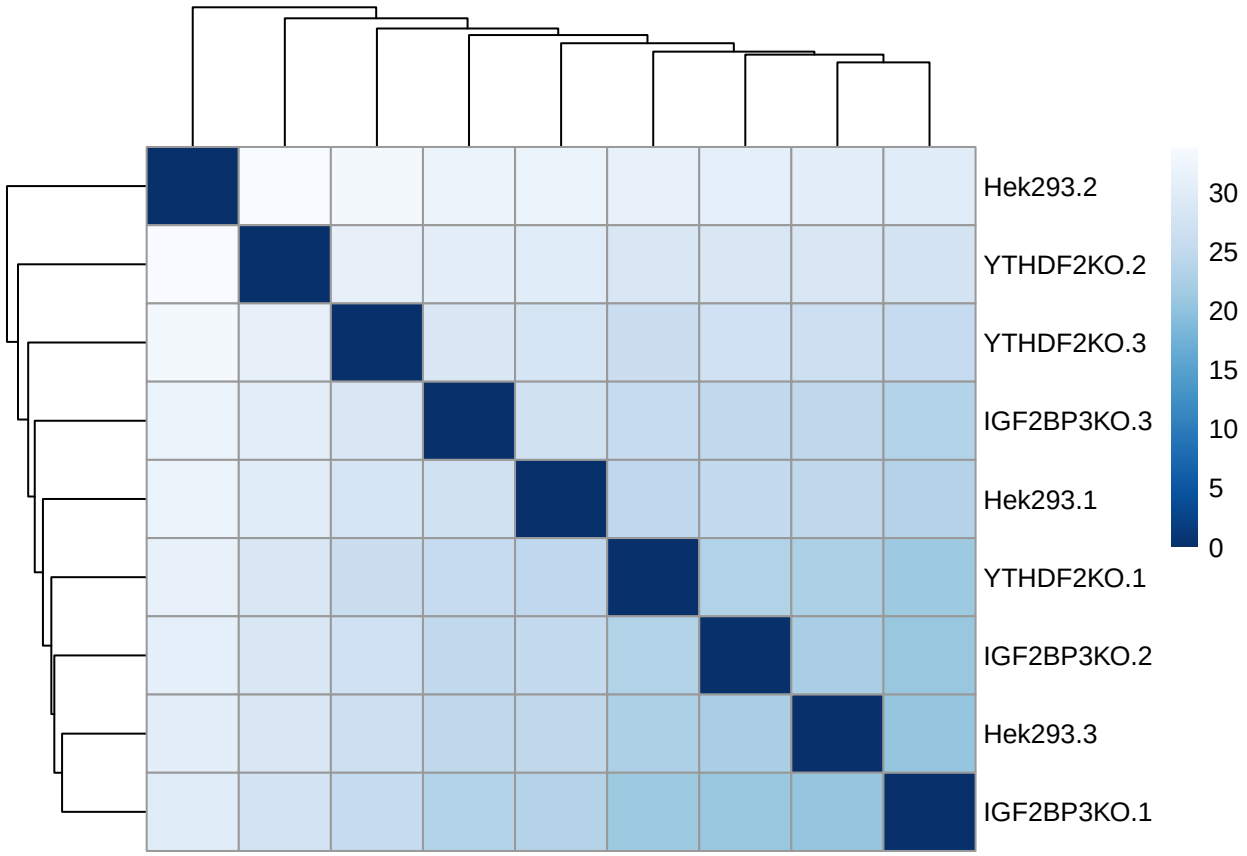



```
# Normalize counts
vsd.one <- vst(dds.all, blind=FALSE)
rlog.one <- rlog(dds.all, blind=FALSE)
```

```
## -- note: fitType='parametric', but the dispersion trend was not well captured by the
##    function: y = a/x + b, and a local regression fit was automatically substituted.
##    specify fitType='local' or 'mean' to avoid this message next time.
```

```
# Keep genes with at least 10 reads total across samples
keep <- rowSums(counts(dds.all)) >= 10
dds.all <- dds.all[keep,]

# Calculate distances between samples
sampleDists <- dist(t(assay(vsd.one)))

# Plot inter-sample distances
old.par <- par(no.readonly=T)

sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- paste(rlog.one$sample_id)
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
p.pheatmap <- pheatmap(sampleDistMatrix,
```

```
         clustering_distance_rows=sampleDists,
         clustering_distance_cols=sampleDists,
         col=colors)
```



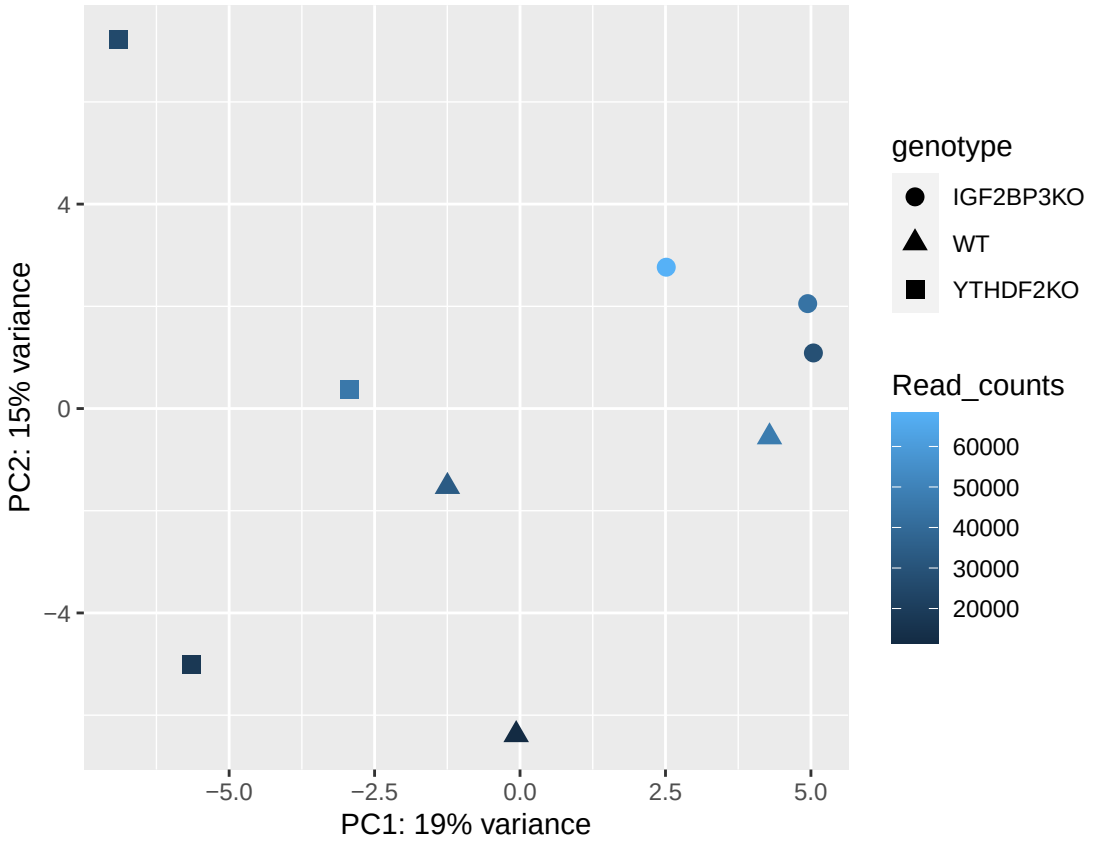

```
ggsave2(filename = "unsupervised_clustering_rnaseq_profile_20plus_reads.pdf", plot = p.pheatmap, path =
```

```
## Saving 6.5 x 4.5 in image
```

```
# PCA
pcaData <- plotPCA(rlog.one, intgroup=c("genotype", "Read_counts"), returnData=TRUE)
percentVar <- round(100 * attr(pcaData, "percentVar"))
y.coords = c(min(pcaData$PC1, pcaData$PC2), max(pcaData$PC1, pcaData$PC2))
x.coords = y.coords
p1 <- ggplot(pcaData, aes(PC1, PC2, shape=genotype, color=Read_counts )) +
  geom_point(size=3) + #scale_color_lancet() +
  xlab(paste0("PC1: ",percentVar[1],"% variance")) +
  ylab(paste0("PC2: ",percentVar[2],"% variance")) +
  coord_fixed(ratio = (max(pcaData$PC1)-min(pcaData$PC1))/(max(pcaData$PC2)-min(pcaData$PC2)))

ggsave("Plots/pca_dataset_1_Induc_gt.pdf", plot = p1)
```

```
## Saving 6.5 x 4.5 in image
```

```
p1
```



Samples separate mainly by sequencing depth (Low <= 5x10e5 reads). Hence, it is important to control by sequencing depth during DE analysis.

resultsNames(dds)

Filtering out poorly-expressed genes (less than 10 reads across all samples)

```
# Keep genes with at least 10 reads total across samples
keep <- rowSums(counts(dds.all)) >= 10
dds.all <- dds.all[keep,]

#dds.rnaseA <- dds.all[ , dds.all$Exp_Group == "RNaseA_exp"]
#dds.rnaseA$Genotype <- droplevels( dds.rnaseA$Genotype)
#dds.rnaseA$Treatment <- droplevels(dds.rnaseA$Treatment)
#dds.rnaseA$Read_depth <- droplevels( dds.rnaseA$Read_depth)

#dds.rnaseH <- dds.all[ , dds.all$Exp_Group == "RNaseH_exp"]
#dds.rnaseH$Genotype <- droplevels(dds.rnaseH$Genotype)
#dds.rnaseH$Treatment <- droplevels(dds.rnaseH$Treatment)
#dds.rnaseH$Read_depth <- droplevels(dds.rnaseH$Read_depth)
```

Using groups instead of interactions

```
# Define function for processing and saving result tables
sort_and_write_res_table <- function(result_table, file_name){
  dir.create(path = "./DE", showWarnings = FALSE)
  # Sort genes by (padj)
  result_table_sorted <- result_table[order(result_table$padj, decreasing = FALSE),]
  # Add gene symbols
  gene_list <- rownames(result_table_sorted)
  symbol_list <- ensembl_to_symbol$gene_name[match(gene_list, ensembl_to_symbol$Ensembl_ID)]
  df <-as.data.frame(cbind(result_table_sorted, Gene_name = symbol_list))

  # Write sorted table to file
  write.table(df, file = paste0("./DE/",file_name,".txt"),
              sep = "\t", col.names=NA)
  return(result_table_sorted)
}



# Calculate DE for WT samples
design(dds.all) <- ~genotype
dds.all$genotype <- relevel(dds.all$genotype, "WT")
dds.all <- DESeq(dds.all)
```

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
```

```
resultsNames(dds.all)
```

```
## [1] "Intercept"                "genotype_IGF2BP3KO_vs_WT"
## [3] "genotype_YTHDF2KO_vs_WT"
```

```
# Using lfcShrink instead of results to reduce high Log2FC bias of genes with low expression
#res_genotype_IGF2BP3KO_vs_WT <- lfcShrink(dds.all, coef = "genotype_IGF2BP3KO_vs_WT", type = "ashr", )
#res_genotype_YTHDF2KO_vs_WT <- lfcShrink(dds.all, coef = "genotype_YTHDF2KO_vs_WT", type = "ashr", )

res_genotype_IGF2BP3KO_vs_WT <- results(dds.all, name = "genotype_IGF2BP3KO_vs_WT")
res_genotype_YTHDF2KO_vs_WT <- results(dds.all, name = "genotype_YTHDF2KO_vs_WT")

summary(res_genotype_IGF2BP3KO_vs_WT, alpha = 0.05)
```

```
##
## out of 7344 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)       : 0, 0%
## LFC < 0 (down)     : 0, 0%
## outliers [1]       : 0, 0%
## low counts [2]     : 0, 0%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
summary(res_genotype_YTHDF2KO_vs_WT, alpha = 0.05)
```

```
##
## out of 7344 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)       : 0, 0%
## LFC < 0 (down)     : 0, 0%
## outliers [1]       : 0, 0%
## low counts [2]     : 0, 0%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
# Sort results by Log2FC
res_genotype_IGF2BP3KO_vs_WT_sorted <- sort_and_write_res_table(res_genotype_IGF2BP3KO_vs_WT, "DE_IGF2BP
res_genotype_YTHDF2KO_vs_WT_sorted <- sort_and_write_res_table(res_genotype_YTHDF2KO_vs_WT, "DE_YTHDF2KO

# Save sorted files as a list
DE_results = list()
DE_results[["IGF2BP3KO_vs_WT"]]  <- res_genotype_IGF2BP3KO_vs_WT_sorted
DE_results[["YTHDF2KO_vs_WT"]] <- res_genotype_YTHDF2KO_vs_WT_sorted
```

```
sessionInfo()
```

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] grid      stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] ggraph_2.0.5                broom_1.0.0
```

```
##  [3] ggupset_0.3.0              ReactomePA_1.38.0
##  [5] enrichplot_1.14.2          cowplot_1.1.1
##  [7] msigdbr_7.5.1              RColorBrewer_1.1-3
##  [9] viridis_0.6.2              viridisLite_0.4.0
## [11] ggsci_2.9                  GOSemSim_2.20.0
## [13] clusterProfiler_4.2.2      VennDiagram_1.7.3
## [15] futile.logger_1.4.3        pcaExplorer_2.20.2
## [17] forcats_0.5.1              dplyr_1.0.9
## [19] purrr_0.3.4                readr_2.1.2
## [21] tidyr_1.2.0                tibble_3.1.7
## [23] tidyverse_1.3.1            biomaRt_2.50.3
## [25] stringr_1.4.0              DESeq2_1.34.0
## [27] SummarizedExperiment_1.24.0 MatrixGenerics_1.6.0
## [29] matrixStats_0.62.0         GenomicRanges_1.46.1
## [31] GenomeInfoDb_1.30.1        ggpubr_0.4.0
## [33] EnhancedVolcano_1.12.0     ggrepel_0.9.1
## [35] ggplot2_3.3.6              pheatmap_1.0.12
## [37] org.Hs.eg.db_3.14.0        AnnotationDbi_1.56.2
## [39] IRanges_2.28.0             S4Vectors_0.32.4
## [41] Biobase_2.54.0             BiocGenerics_0.40.0
##
## loaded via a namespace (and not attached):
##   [1] rappdirs_0.3.3         SparseM_1.81           AnnotationForge_1.36.0
##   [4] ragg_1.2.2             pkgmaker_0.32.2        bit64_4.0.5
##   [7] knitr_1.39             DelayedArray_0.20.0    data.table_1.14.2
##  [10] KEGGREST_1.34.0        RCurl_1.98-1.7         doParallel_1.0.17
##  [13] generics_0.1.3         lambda.r_1.2.4         RSQLite_2.2.14
##  [16] shadowtext_0.1.2       bit_4.0.4              tzdb_0.3.0
##  [19] webshot_0.5.3          xml2_1.3.3             lubridate_1.8.0
##  [22] httpuv_1.6.5           assertthat_0.2.1       xfun_0.31
##  [25] hms_1.1.1              babelgene_22.3         evaluate_0.15
##  [28] promises_1.2.0.1       TSP_1.2-0              fansi_1.0.3
##  [31] progress_1.2.2         dendextend_1.16.0      dbplyr_2.2.1
##  [34] readxl_1.4.0           Rgraphviz_2.38.0       igraph_1.3.2
##  [37] DBI_1.1.3              geneplotter_1.72.0     htmlwidgets_1.5.4
##  [40] ellipsis_0.3.2         crosstalk_1.2.0        backports_1.4.1
##  [43] annotate_1.72.0        gridBase_0.4-7         vctrs_0.4.1
##  [46] abind_1.4-5            cachem_1.0.6           withr_2.5.0
##  [49] ggforce_0.3.3          checkmate_2.1.0        treeio_1.18.1
##  [52] prettyunits_1.1.1      cluster_2.1.2          DOSE_3.23.2
##  [55] ape_5.6-2              lazyeval_0.2.2         crayon_1.5.1
##  [58] genefilter_1.76.0      labeling_0.4.2         pkgconfig_2.0.3
##  [61] tweenr_1.0.2           nlme_3.1-152           vipor_0.4.5
##  [64] seriation_1.3.5        rlang_1.0.3            lifecycle_1.0.1
##  [67] downloader_0.4         registry_0.5-1         filelock_1.0.2
##  [70] extrafontdb_1.0        BiocFileCache_2.2.1    GOstats_2.60.0
##  [73] modelr_0.1.8           ggrastr_1.0.1          cellranger_1.1.0
##  [76] polyclip_1.10-0        graph_1.72.0           rngtools_1.5.2
##  [79] aplot_0.1.6            Matrix_1.3-4           carData_3.0-5
##  [82] reprex_2.0.1           base64enc_0.1-3        beeswarm_0.4.0
##  [85] png_0.1-7              bitops_1.0-7           shinydashboard_0.7.2
##  [88] KernSmooth_2.23-20     Biostrings_2.62.0      blob_1.2.3
##  [91] qvalue_2.26.0          gridGraphics_0.5-1     rstatix_0.7.0
##  [94] shinyAce_0.4.2         ggsignif_0.6.3         reactome.db_1.77.0
```

```
##  [97] scales_1.2.0          graphite_1.40.0       memoise_2.0.1
## [100] GSEABase_1.56.0       magrittr_2.0.3        plyr_1.8.7
## [103] zlibbioc_1.40.0       threejs_0.3.3         scatterpie_0.1.7
## [106] compiler_4.1.1        ash_1.0-15            cli_3.3.0
## [109] XVector_0.34.0        Category_2.60.0       patchwork_1.1.1
## [112] formatR_1.12          MASS_7.3-54           tidyselect_1.1.2
## [115] stringi_1.7.6         textshaping_0.3.6     shinyBS_0.61.1
## [118] highr_0.9             proj4_1.0-11          yaml_2.3.5
## [121] locfit_1.5-9.5        fastmatch_1.1-3       tools_4.1.1
## [124] parallel_4.1.1        rstudioapi_0.13       foreach_1.5.2
## [127] gridExtra_2.3         farver_2.1.1          digest_0.6.29
## [130] shiny_1.7.1           Rcpp_1.0.9            car_3.1-0
## [133] ggalt_0.4.0           later_1.3.0           httr_1.4.3
## [136] colorspace_2.0-3      rvest_1.0.2           XML_3.99-0.10
## [139] fs_1.5.2              topGO_2.46.0          splines_4.1.1
## [142] yulab.utils_0.0.5     RBGL_1.70.0           tidytree_0.3.9
## [145] graphlayouts_0.8.0    ggplotify_0.1.0       systemfonts_1.0.4
## [148] plotly_4.10.0         xtable_1.8-4          ggtree_3.2.1
## [151] jsonlite_1.8.0        futile.options_1.0.1  heatmaply_1.3.0
## [154] tidygraph_1.2.1       ggfun_0.0.6           R6_2.5.1
## [157] pillar_1.7.0          htmltools_0.5.2       mime_0.12
## [160] NMF_0.24.0            glue_1.6.2            fastmap_1.1.0
## [163] DT_0.23               BiocParallel_1.28.3   codetools_0.2-18
## [166] maps_3.4.0            fgsea_1.20.0          utf8_1.2.2
## [169] lattice_0.20-44       curl_4.3.2            ggbeeswarm_0.6.0
## [172] GO.db_3.14.0          Rttf2pt1_1.3.10       survival_3.2-11
## [175] limma_3.50.3          rmarkdown_2.14        munsell_0.5.0
## [178] DO.db_2.9             GenomeInfoDbData_1.2.7 iterators_1.0.14
## [181] haven_2.5.0           reshape2_1.4.4        gtable_0.3.0
## [184] extrafont_0.18
```