Title Goes Here

Jane Doe

September 29, 2025

Abstract

Text of abstract

Keywords: keyword 1; keyword 2; keyword 3

Highlights: These are the highlights.

library(DESeq2)  
library(ggplot2)  
library(ggrepel)  
library(gprofiler2)  
library(openxlsx)  
library(dplyr)  
  
set\_base\_url("https://biit.cs.ut.ee/gprofiler\_archive3/e104\_eg51\_p15")  
  
volcano\_plot <- function(exp, down\_color, up\_color, title){  
 colnames(exp)[grep(pattern = "log2FoldChange",x = colnames(exp))] <- "log2FoldChange"  
 colnames(exp)[grep(pattern = "padj",x = colnames(exp))] <- "padj"  
   
 exp$direction <- "No-significant"  
   
 exp$direction[which(exp$padj < 0.05 & exp$log2FoldChange > 0)] <- "Up-regulated"  
 exp$direction[which(exp$padj < 0.05 & exp$log2FoldChange < 0)] <- "Down-regulated"  
   
 exp <- exp[order(exp$padj),]  
   
 exp$threshold <- c(rep(TRUE,10),rep(FALSE,nrow(exp) - 10))  
 exp$direction <- as.factor(exp$direction)  
 result <- exp[which(!is.na(exp$padj)),]  
 result <- result[order(result$threshold,decreasing = TRUE),]  
   
 ggplot(result) +  
 geom\_point(aes(x=log2FoldChange, y=-log10(padj),colour=direction)) +   
 scale\_color\_manual(values=c(down\_color,"grey",up\_color)) + ggtitle(title) +  
 geom\_text\_repel(data =result[c(1:10),],   
 mapping =aes(x= log2FoldChange, y = -log10(padj), label = result$name[c(1:10)] ), size = 1.5) +  
 xlab("log2 fold-change") +   
 ylab("-log10 adj. p-value") +  
 geom\_hline(yintercept=-log10(0.05), linetype="dashed", color = "red") + theme\_classic() + theme(legend.position = "none",   
 text = element\_text(size = 8),  
 plot.title = element\_text(size=6))   
   
}  
loc\_val <- function(mean\_FC,SD){  
 result <- c()  
 for(i in seq(length(mean\_FC))){  
 if(SD[i] == 0){SD[i] <- min(SD[which(SD != 0)])}  
   
 if(mean\_FC[i] < 0){  
 result <- c(result,mean\_FC[i]-SD[i]-(min(SD[i])/2))  
 }  
 if(mean\_FC[i] > 0){  
 result <- c(result,mean\_FC[i]+SD[i]+(min(SD[i])/2))  
 }  
 if(mean\_FC[i] == 0){  
 result <- c(result,NA)  
 }  
 }  
 return(result)  
}  
enrichment\_profile = function(pathways, DEG, color\_down, color\_up, category){  
 # data is the dataframe containing only significant genes for a specific cell type  
 # pathway is a df of pathway ids/go ids that are statistically enriched from geneprofile  
   
 DEG$name <- rownames(DEG)  
   
 colnames(DEG)[grep(pattern = "log2fc",x = colnames(DEG))] <- "log2FoldChange"  
 colnames(DEG)[grep(pattern = "pvalue",x = colnames(DEG))] <- "padj"  
 colnames(DEG)[grep(pattern = "name",x = colnames(DEG))] <- "ensembl\_gene\_id"  
   
 to.plot <- data.frame()  
 for(i in 1:nrow(pathways)){  
 res = gost(query = pathways[i,"term\_id"], organism = "hsapiens")  
 gene\_set = res$meta$genes\_metadata$query$query\_1$ensgs  
   
 to.plot <- rbind(to.plot,  
 data.frame(  
 Name=pathways[i,"term\_name"],  
 mean\_FC=c(mean(DEG[DEG$ensembl\_gene\_id %in% gene\_set,]$log2FoldChange[DEG[DEG$ensembl\_gene\_id %in% gene\_set,]$log2FoldChange > 0]),  
 mean(DEG[DEG$ensembl\_gene\_id %in% gene\_set,]$log2FoldChange[DEG[DEG$ensembl\_gene\_id %in% gene\_set,]$log2FoldChange < 0])),  
 Type=c("up-regulated","down-regulated"),  
 Set\_size = c(length(DEG[DEG$ensembl\_gene\_id %in% gene\_set,]$log2FoldChange[DEG[DEG$ensembl\_gene\_id %in% gene\_set,]$log2FoldChange > 0]),  
 length(DEG[DEG$ensembl\_gene\_id %in% gene\_set,]$log2FoldChange[DEG[DEG$ensembl\_gene\_id %in% gene\_set,]$log2FoldChange < 0])),  
 SD=c(sd(DEG[DEG$ensembl\_gene\_id %in% gene\_set,]$log2FoldChange[DEG[DEG$ensembl\_gene\_id %in% gene\_set,]$log2FoldChange > 0]),  
 sd(DEG[DEG$ensembl\_gene\_id %in% gene\_set,]$log2FoldChange[DEG[DEG$ensembl\_gene\_id %in% gene\_set,]$log2FoldChange < 0]))  
 ))  
 }  
   
 to.plot[is.nan(to.plot$mean\_FC),"mean\_FC"] <- 0  
 to.plot[is.na(to.plot$SD),"SD"] <- 0  
   
 to.plot$Name[which(to.plot$Name == "Regulation of Insulin-like Growth Factor (IGF) transport and uptake by Insulin-like Growth Factor Binding Proteins (IGFBPs)")] <- "Regulation of Insulin-like Growth Factor\n (IGF) transport and uptake by Insulin-like\n Growth Factor Binding Proteins (IGFBPs)"  
   
 to.plot$Name[which(to.plot$Name == "Assembly of collagen fibrils and other multimeric structures")] <- "Assembly of collagen fibrils\n and other multimeric structures"  
   
   
 p <- ggplot(to.plot, aes(x = mean\_FC, y = reorder(Name, abs(mean\_FC)), fill = Type, label = Set\_size)) +   
 scale\_fill\_manual(name = "", values = c("down-regulated" = color\_down, "up-regulated" = color\_up)) +   
 ylab(category) + xlab("Mean logFC") +   
 theme(axis.line = element\_blank(),  
 panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank(),  
 panel.background = element\_blank()) + theme(text = element\_text(size = 6)) +   
 theme(legend.position = "none") +  
 geom\_bar(stat = "identity") + geom\_text(aes(label = Set\_size, x = loc\_val(mean\_FC,SD) ),size = 1.5) +  
 geom\_linerange(aes(xmin=ifelse(mean\_FC < 0,mean\_FC - SD, mean\_FC), xmax=ifelse(mean\_FC > 0,mean\_FC + SD, mean\_FC) ))  
   
 return(p)  
}  
  
raw\_data <- "/workspaces/phd\_dissertation/PIPE/analysis/data/raw\_data/"  
first <- read.xlsx(paste0(raw\_data,"231203\_raw\_data.xlsx"),sheet = 1)

# Mono-culture

## Endothelial 21% O2. Normal shear stress

counts <- first[,grep(pattern = 'raw\_E21\_BR|raw\_DEN21\_BR',x = colnames(first))]  
rownames(counts) <- first$id  
  
meta <- data.frame(experiment = as.factor(gsub(pattern = "\\\_BR[1-3]",replacement = "",x = colnames(counts))))  
rownames(meta) <- colnames(counts)  
  
dds <- DESeqDataSetFromMatrix(countData = counts,  
 colData = meta,  
 design = ~experiment)  
  
keep <- rowSums(counts(dds)) >= 10  
dds <- dds[keep,]  
dds$experiment <- relevel(dds$experiment, ref = "raw\_E21")  
dds <- DESeq(dds)  
resultsNames(dds)

[1] "Intercept" "experiment\_raw\_DEN21\_vs\_raw\_E21"

resLFC <- lfcShrink(dds, coef=2, type="apeglm")  
resLFC <- as.data.frame(resLFC) %>% mutate(name = first$name[which(first$id %in% rownames(resLFC))], .before=1)  
cat("number of significant genes: ",nrow(resLFC[which(resLFC$padj < 0.05),]))

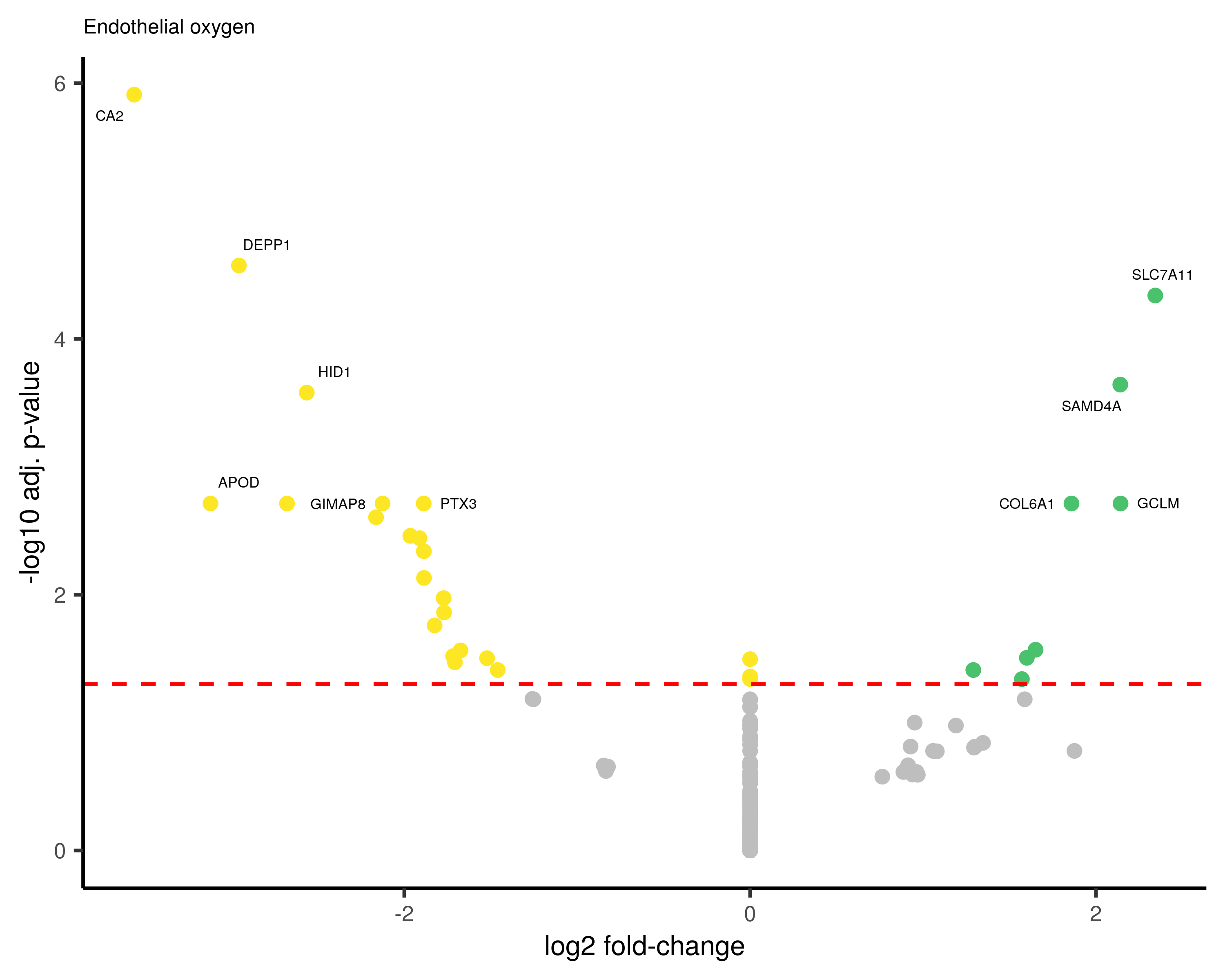
number of significant genes: 31

head(resLFC[order(resLFC$padj),])

name baseMean log2FoldChange lfcSE pvalue  
ENSG00000104267 CA2 25.81729 -3.561646 0.6135713 3.242714e-10  
ENSG00000165507 DEPP1 25.27896 -2.955909 0.5788937 1.405284e-08  
ENSG00000151012 SLC7A11 55.37205 2.342875 0.4686745 3.618823e-08  
ENSG00000020577 SAMD4A 34.05854 2.140450 0.4688876 2.399889e-07  
ENSG00000167861 HID1 31.14038 -2.563661 0.5771215 3.464114e-07  
ENSG00000189058 APOD 29.30900 -3.120242 0.8066631 3.843185e-06  
 padj  
ENSG00000104267 1.229961e-06  
ENSG00000165507 2.665121e-05  
ENSG00000151012 4.575398e-05  
ENSG00000020577 2.275695e-04  
ENSG00000167861 2.627877e-04  
ENSG00000189058 1.934964e-03

### Volcano plot

volcano\_plot(exp = resLFC,down\_color = "#FDE725FF",up\_color = "#4AC16DFF", title="Endothelial oxygen")



# Session info

sessionInfo()

R version 4.2.3 (2023-03-15)  
Platform: x86\_64-pc-linux-gnu (64-bit)  
Running under: Ubuntu 22.04.4 LTS  
  
Matrix products: default  
BLAS: /usr/lib/x86\_64-linux-gnu/openblas-pthread/libblas.so.3  
LAPACK: /usr/lib/x86\_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.20.so  
  
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] dplyr\_1.1.1 openxlsx\_4.2.5.2   
 [3] gprofiler2\_0.2.1 ggrepel\_0.9.3   
 [5] ggplot2\_3.4.2 DESeq2\_1.36.0   
 [7] SummarizedExperiment\_1.26.1 Biobase\_2.56.0   
 [9] MatrixGenerics\_1.8.1 matrixStats\_0.63.0   
[11] GenomicRanges\_1.48.0 GenomeInfoDb\_1.32.4   
[13] IRanges\_2.30.1 S4Vectors\_0.34.0   
[15] BiocGenerics\_0.42.0   
  
loaded via a namespace (and not attached):  
 [1] bitops\_1.0-7 bit64\_4.0.5 RColorBrewer\_1.1-3   
 [4] httr\_1.4.5 numDeriv\_2016.8-1.1 tools\_4.2.3   
 [7] utf8\_1.2.3 R6\_2.5.1 DBI\_1.1.3   
[10] lazyeval\_0.2.2 colorspace\_2.1-0 apeglm\_1.27.0   
[13] withr\_2.5.0 tidyselect\_1.2.0 bit\_4.0.5   
[16] compiler\_4.2.3 cli\_3.6.1 DelayedArray\_0.22.0   
[19] plotly\_4.10.1 labeling\_0.4.2 scales\_1.2.1   
[22] mvtnorm\_1.1-3 genefilter\_1.78.0 digest\_0.6.31   
[25] rmarkdown\_2.21 XVector\_0.36.0 pkgconfig\_2.0.3   
[28] htmltools\_0.5.5 fastmap\_1.1.1 bbmle\_1.0.25   
[31] htmlwidgets\_1.6.2 rlang\_1.1.0 RSQLite\_2.3.1   
[34] generics\_0.1.3 farver\_2.1.1 jsonlite\_1.8.4   
[37] BiocParallel\_1.30.4 zip\_2.3.0 RCurl\_1.98-1.12   
[40] magrittr\_2.0.3 GenomeInfoDbData\_1.2.8 Matrix\_1.5-4   
[43] Rcpp\_1.0.10 munsell\_0.5.0 fansi\_1.0.4   
[46] lifecycle\_1.0.3 stringi\_1.7.12 yaml\_2.3.7   
[49] MASS\_7.3-58.3 zlibbioc\_1.42.0 plyr\_1.8.8   
[52] grid\_4.2.3 blob\_1.2.4 parallel\_4.2.3   
[55] bdsmatrix\_1.3-6 crayon\_1.5.2 lattice\_0.21-8   
[58] Biostrings\_2.64.1 splines\_4.2.3 annotate\_1.74.0   
[61] KEGGREST\_1.36.3 locfit\_1.5-9.7 knitr\_1.42   
[64] pillar\_1.9.0 geneplotter\_1.74.0 codetools\_0.2-19   
[67] XML\_3.99-0.14 glue\_1.6.2 evaluate\_0.20   
[70] data.table\_1.14.8 png\_0.1-8 vctrs\_0.6.2   
[73] gtable\_0.3.3 purrr\_1.0.1 tidyr\_1.3.0   
[76] cachem\_1.0.7 emdbook\_1.3.12 xfun\_0.38   
[79] xtable\_1.8-4 coda\_0.19-4 survival\_3.5-5   
[82] viridisLite\_0.4.1 tibble\_3.2.1 AnnotationDbi\_1.58.0   
[85] memoise\_2.0.1