RNA splicing analysis after the knockdown of HNRNPC, UPF1, and CSDE1 after 1h EGF signaling

Luca Ducoli

02 July, 2024

This is the pipeline used to analyze the splicing results after the knockdown of HNRNPC, CSDE1, and UPF1 in A431 after 1h EGF treatment. This experiment was performed in two replicates in A431 cells.

1. Visualization of splicing results

Splicing analysis was performed using rMATs (PMID:25480548).

```
#Needed libraries
library(ggplot2)
library(viridis)
library(hrbrthemes)
library(tidyverse)
library(tidyr)
library(stringr)
library(pheatmap)
library(UpSetR)
library(gridExtra)
library(grid)
library(RColorBrewer)
library(reshape2)
library(psych)
library(factoextra)
library(ggpubr)
library(ggrepel)
library(gprofiler2)
```

```
#Directory with samples
dir <- "/Users/lducoli/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/1_RNA-Seq
    _splicing/2_Splicing_analysis/1_rmats/2_replicates"
sample <- list.files(dir)

vis_rmats <- function(AS_event, type) {
  # AS_event <- "A3SS"
  # type <- "flanking"

  files <- file.path(dir, sample, paste(AS_event, "MATS.JCEC.txt", sep = "."))
  names(files) <- paste0(sample)
  all(file.exists(files))

  for (i in 1:length(files)) assign(sample[i], read.delim(files[i]))

  #reorganize the table
```

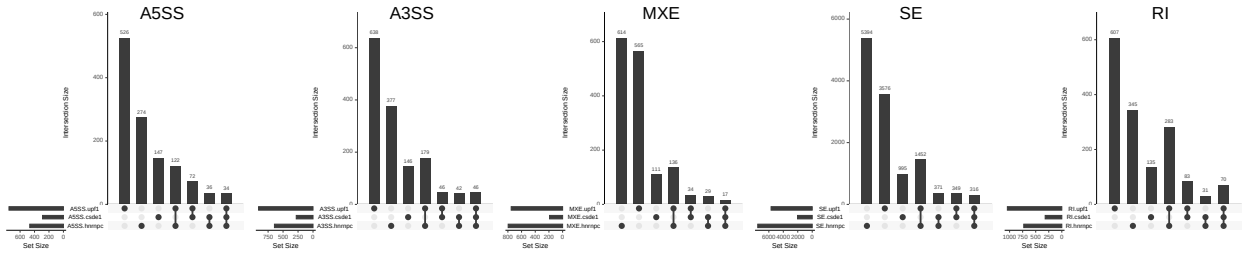
```
csde1_AS <- csde1_vs_ctrl %>% dplyr::select(-c(ID.1)) %>% unite( "GeneName_ID", c(geneSymbol,ID
  ), sep = "_", remove = FALSE) %>% unite( "region", GeneID:paste(type, "EE", sep = ""), sep =
  "_", remove = TRUE) %>% tidyr::extract(IncLevel1, c("csde1_rep1", "csde1_rep2"), "([^)]+)
  ,([^)]+)") %>% tidyr::extract(IncLevel2, c("Ctrl_rep1", "Ctrl_rep2"), "([^)]+),([^)]+)") %>%
  dplyr::rename(csde1_PValue = PValue, csde1_FDR = FDR, csde1_dPSI = IncLevelDifference)
hnrnpc_AS <- hnrnpc_vs_ctrl %>% dplyr::select(-c(ID.1)) %>% unite( "GeneName_ID", c(geneSymbol,
  ID), sep = "_", remove = FALSE) %>% unite( "region", GeneID:paste(type, "EE", sep = ""), sep
  = "_", remove = TRUE) %>% tidyr::extract(IncLevel1, c("hnrnpc_rep1", "hnrnpc_rep2"), "([^)]+)
  ,([^)]+)") %>% tidyr::extract(IncLevel2, c("Ctrl_rep1", "Ctrl_rep2"), "([^)]+),([^)]+)") %>%
  dplyr::rename(hnrnpc_PValue = PValue, hnrnpc_FDR = FDR, hnrnpc_dPSI = IncLevelDifference)
upf1_AS <- upf1_vs_ctrl %>% dplyr::select(-c(ID.1)) %>% unite( "GeneName_ID", c(geneSymbol,ID),
   sep = "_", remove = FALSE) %>% unite( "region", GeneID:paste(type, "EE", sep = ""), sep = "_
  ", remove = TRUE) %>% tidyr::extract(IncLevel1, c("upf1_rep1", "upf1_rep2"), "([^)]+),([^)]+)
  ") %>% tidyr::extract(IncLevel2, c("Ctrl_rep1", "Ctrl_rep2"), "([^)]+),([^)]+)") %>% dplyr::
  rename(upf1_PValue = PValue, upf1_FDR = FDR, upf1_dPSI = IncLevelDifference)

#merge the different tables
AS <- merge(csde1_AS %>% dplyr::select(region, Ctrl_rep1, Ctrl_rep2, csde1_dPSI, csde1_PValue,
  csde1_FDR, csde1_rep1, csde1_rep2), hnrnpc_AS %>% dplyr::select(region, hnrnpc_dPSI, hnrnpc_
  PValue, hnrnpc_FDR, hnrnpc_rep1, hnrnpc_rep2), by="region", all.x=TRUE, all.y=TRUE)
AS <- merge(AS, upf1_AS %>% dplyr::select(region, upf1_dPSI, upf1_PValue, upf1_FDR, upf1_rep1,
  upf1_rep2), by="region", all.x=TRUE, all.y=TRUE)
  write.table(AS, file = paste("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_
  EGF/1_RNA-Seq_splicing/2_Splicing_analysis/2_Correlation/", AS_event,"_events_all.txt", sep =
   ""), quote = F, sep = "\t", row.names = F)

#Filter sgnificant AS events
AS.csde1 <- subset(AS, csde1_FDR < 0.05)
AS.csde1.2 <- merge(csde1_AS[c(2:5)], AS.csde1, by = "region")
AS.hnrnpc <- subset(AS, hnrnpc_FDR < 0.05)
AS.hnrnpc.2 <- merge(hnrnpc_AS[c(2:5)], AS.hnrnpc, by = "region")
AS.upf1 <- subset(AS, upf1_FDR < 0.05)
AS.upf1.2 <- merge(upf1_AS[c(2:5)], AS.upf1, by = "region")

AS_final <- unique(rbind(AS.csde1, AS.hnrnpc, AS.upf1))

#Upset plot
lt.tsk = list(unique(AS.hnrnpc$region), unique(AS.csde1$region), unique(AS.upf1$region))
names(lt.tsk) <- c(paste(AS_event, "hnrnpc", sep = "."), paste(AS_event, "csde1", sep = "."),
  paste(AS_event, "upf1", sep = "."))

upset <- upset(fromList(lt.tsk),
  sets = c(names(lt.tsk)[1],names(lt.tsk)[2], names(lt.tsk)[3]),
  mb.ratio = c(0.8, 0.2),
  number.angles = 0,
  text.scale = 1,
  point.size = 3,
  line.size = 1,
  keep.order = TRUE
)

  # Function to create the binary table
overlap <- function (input) {
  elements <- unique(unlist(input))
  data <- unlist(lapply(input, function(x) {
    x <- as.vector(match(elements, x))
  }))
  data[is.na(data)] <- as.integer(0)
  data[data != 0] <- as.integer(1)
  data <- data.frame(matrix(data, ncol = length(input), byrow = F))
  data <- data[which(rowSums(data) != 0), ]
  names(data) <- names(input)
  row.names(data) <- elements
  return(data)
}

# Binary table with colnames:
sign.genes <- overlap(lt.tsk)
```

```
  sign.genes$region <- rownames(sign.genes)
  rownames(sign.genes) <- NULL

  #Heatmap
  data_columns <- grep("dPSI", colnames(AS_final))
  data <- as.matrix(AS_final[,data_columns])
  data[data == "NA"] <- 0
  data[is.na(data)] <- 0
  rownames(data) <- AS_final$region
  all <- data
  cormat <- round(cor(data),2)

  reorder_cormat <- function(cormat){
  dd <- as.dist((1-cormat)/2)
  hc <- hclust(dd)
  cormat <-cormat[hc$order, hc$order]
  }

  cormat <- reorder_cormat(cormat)

  my.breaks <- c(seq(0.4, 0.6, by=0.01))
  my.colors <- colorRampPalette(colors = c("#FFFFFF", "#FFF5F5", "#FFECEC", "#FDDBC7", "#F4A582",
      "#D6604D", "#B2182B"))(length(my.breaks))

  heatmap <- pheatmap(
  mat               = cormat,
  cellwidth = 40,
  cellheight = 40,
  color = my.colors,
  breaks = my.breaks,
  show_colnames     = TRUE,
  show_rownames     = TRUE,
  drop_levels       = TRUE,
  fontsize          = 10,
  cluster_rows      = hclust(as.dist((1-cormat)/2)),
  cluster_cols      = hclust(as.dist((1-cormat)/2)),
  angle_col = 45,
  display_numbers = TRUE,
  main = AS_event,
  silent = TRUE
  )

  psi <- as.data.frame(all)
  psi$region <- rownames(psi)
  sign.genes <- merge(sign.genes, psi, by = "region")

return(list(upset = upset, heatmap = heatmap, csde1 = AS.csde1.2, hnrnpc = AS.hnrnpc.2, upf1 = AS
    .upf1.2, all = all, AS_all = AS, overlap = sign.genes))
}

A3SS <- vis_rmats("A3SS", "flanking")
A5SS <- vis_rmats("A5SS", "flanking")
MXE <- vis_rmats("MXE", "downstream")
SE <- vis_rmats("SE", "downstream")
RI <- vis_rmats("RI", "downstream")

A3SS$overlap$AS <- "A3SS"
colnames(A3SS$overlap) <-  sub("A3SS.", "", colnames(A3SS$overlap))
A5SS$overlap$AS <- "A5SS"
colnames(A5SS$overlap) <-  sub("A5SS.", "", colnames(A5SS$overlap))
MXE$overlap$AS <- "MXE"
colnames(MXE$overlap) <-  sub("MXE.", "", colnames(MXE$overlap))
SE$overlap$AS <- "SE"
colnames(SE$overlap) <-  sub("SE.", "", colnames(SE$overlap))
RI$overlap$AS <- "RI"
colnames(RI$overlap) <-  sub("RI.", "", colnames(RI$overlap))

All_AS_overlap <- rbind(A5SS$overlap, A3SS$overlap, MXE$overlap, SE$overlap, RI$overlap)
```

```
write.table(All_AS_overlap, "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/1
    _RNA-Seq_splicing/2_Splicing_analysis/2_Correlation/Sign_AS_events_all.txt", quote = F, sep =
    "\t", row.names = F)

plot.ls <- list(A5SS = A5SS$upset, A3SS = A3SS$upset, MXE = MXE$upset, SE = SE$upset, RI = RI$
    upset)

for (v in names(plot.ls)) {
  print(plot.ls[[v]])
  grid.text(v, x = 0.65, y=0.97, gp = gpar(fontsize = 20))
  grid.edit('arrange', name = v)
  vp <- grid.grab()
  plot.ls[[v]] <- vp
}
```

Upsetplot

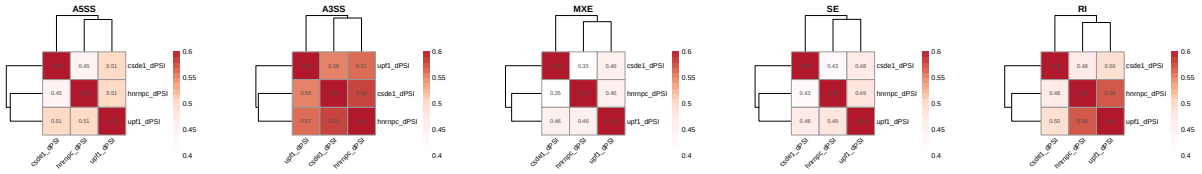
```
do.call("grid.arrange", c(plot.ls, ncol=5))
```




```
# Save the plot as pdf
pdf("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/1_RNA-Seq_splicing/2_
    Splicing_analysis/2_Correlation/Upsetplot_rmats_plot.pdf", height = 4, width = 20)
do.call("grid.arrange", c(plot.ls, ncol=5))
dev.off()
```

Correlation plot

```
hm.ls <- list(A5SS = A5SS$heatmap$gtable, A3SS = A3SS$heatmap$gtable, MXE = MXE$heatmap$gtable,
    SE = SE$heatmap$gtable, RI = RI$heatmap$gtable)
do.call("grid.arrange", c(hm.ls, ncol=5))
```

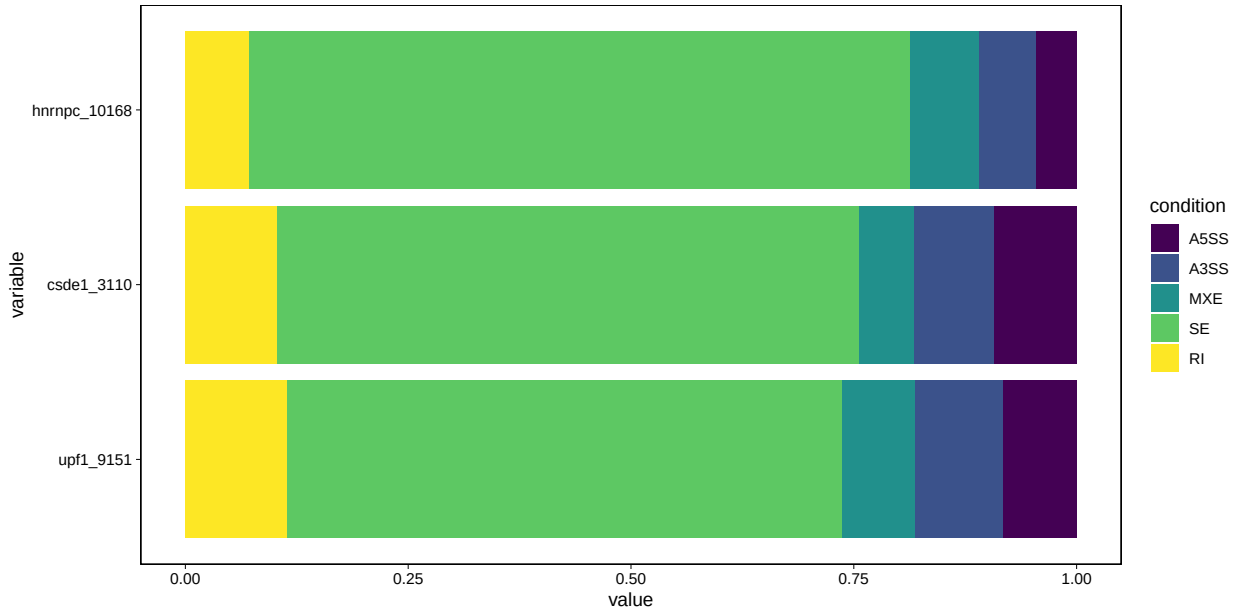



```
# Save the plot as pdf
pdf("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/1_RNA-Seq_splicing/2_
    Splicing_analysis/2_Correlation/Corrheatmap_rmats_plot.pdf", height = 4, width = 20)
```

```
do.call("grid.arrange", c(hm.ls, ncol=5))
dev.off()
```

Barplot of splicing events

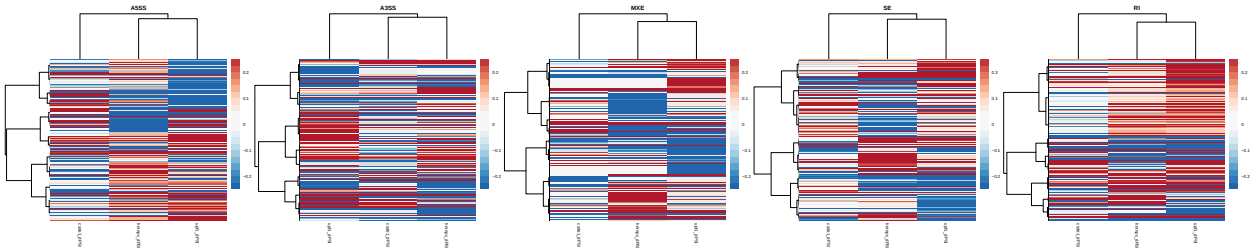
```
#Generate barplot of splicing events
stackedplot <- data.frame(csde1 = c(length(rownames(A5SS$csde1)),length(rownames(A3SS$csde1)),
    length(rownames(MXE$csde1)),length(rownames(SE$csde1)),length(rownames(RI$csde1))),
           hnrnpc = c(length(rownames(A5SS$hnrnpc)),length(rownames(A3SS$hnrnpc)),length(rownames
    (MXE$hnrnpc)),length(rownames(SE$hnrnpc)),length(rownames(RI$hnrnpc))),
           upf1 = c(length(rownames(A5SS$upf1)),length(rownames(A3SS$upf1)),length(rownames(MXE$
    upf1)),length(rownames(SE$upf1)),length(rownames(RI$upf1))))

Sum <- colSums(stackedplot)

stackedplot <- stackedplot %>% mutate(., condition = c('A5SS', 'A3SS', "MXE", "SE", "RI"))
stackedplot <- reshape2::melt(stackedplot)
stackedplot$condition <- factor(stackedplot$condition, levels = c('A5SS', 'A3SS', "MXE", "SE", "
    RI"))
stackedplot$variable <- factor(stackedplot$variable, levels = c("upf1", "csde1","hnrnpc"))
levels(stackedplot$variable) <- c(paste("upf1", Sum[3], sep = "_"), paste("csde1", Sum[1], sep =
    "_"), paste("hnrnpc", Sum[2], sep = "_"))

# Stacked + percent
stackedbarplot <- ggplot(stackedplot, aes(fill=condition, y=value, x=variable)) +
  geom_bar(position="fill", stat="identity")+
  scale_fill_viridis(discrete = T, option = "D") +
  theme_linedraw() + coord_flip() +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        axis.line = element_blank(),
        plot.title = element_text(hjust = 0.5))
stackedbarplot
```



```
# Save the plot as pdf
pdf("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/1_RNA-Seq_splicing/2_
    Splicing_analysis/2_Correlation/Barplot_rmats_plot.pdf", height = 5, width = 10)
stackedbarplot
dev.off()
```

Heatmap of all the events

```
#Generate heatmap of all AS events
heatmap_all <- function(data, AS_event) {
cormat <- round(cor(data),2)

reorder_cormat <- function(cormat,data){
  dd <- as.dist((1-cormat)/2)
  hc <- hclust(dd)
  data <-data[, hc$order]
}

Table.hm <- reorder_cormat(cormat, data)

my.breaks <- c(seq(-0.25, -0.025, by=0.025), seq(0.025, 0.25, by=0.025))
my.colors <- c(colorRampPalette(colors = c("#2166AC", "#4393C3", "#92C5DE", "#D1E5F0", "#F7F7F7")
    )(length(my.breaks)/2), colorRampPalette(colors = c("#F7F7F7", "#FDDBC7", "#F4A582", "#D6604D
    ", "#B2182B"))(length(my.breaks)/2))

Table.hm2 <- scale(Table.hm, center = TRUE)
dist.all <- as.dist((1-cor(Table.hm2))/2)
hclust.all <- hclust(dist.all, method = "complete")

dist.all2 <- as.dist((1-cor(t(Table.hm2)))/2)
hclust.all2 <- hclust(dist.all2, method = "ward.D2")

heatmap <- pheatmap(
  mat               = Table.hm2,
  na_col = "grey",
  color = my.colors,
  breaks = my.breaks,
  show_colnames     = TRUE,
  show_rownames     = FALSE,
  drop_levels       = TRUE,
  fontsize          = 5,
  cluster_rows      = hclust.all2,
  cluster_cols      = hclust.all,
  scale             = "none",
  # cutree_rows = 5,
  main = AS_event,
  silent = TRUE
)
return(heatmap)
}

A3SS.hm <- heatmap_all(A3SS$all, "A3SS")
A5SS.hm <- heatmap_all(A5SS$all, "A5SS")
MXE.hm <- heatmap_all(MXE$all, "MXE")
SE.hm <- heatmap_all(SE$all, "SE")
RI.hm <- heatmap_all(RI$all, "RI")

hm.all.ls <- list(A5SS = A5SS.hm$gtable, A3SS = A3SS.hm$gtable, MXE = MXE.hm$gtable, SE = SE.hm$
    gtable, RI = RI.hm$gtable)
do.call("grid.arrange", c(hm.all.ls, ncol=5))
```




```
# Save the plot as pdf
pdf("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/1_RNA-Seq_splicing/2_
    Splicing_analysis/2_Correlation/Allheatmap_rmats_plot.pdf", height = 4, width = 20)
do.call("grid.arrange", c(hm.all.ls, ncol=5))
dev.off()
```

All the visualizations were saved as pdf and modified in illustrator.

```
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur ... 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2/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      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] gprofiler2_0.2.3   ggrepel_0.9.5      ggpubr_0.6.0       factoextra_1.0.7
##  [5] psych_2.4.3        reshape2_1.4.4     RColorBrewer_1.1-3 gridExtra_2.3
##  [9] UpSetR_1.4.0       pheatmap_1.0.12    lubridate_1.9.3    forcats_1.0.0
## [13] stringr_1.5.1      dplyr_1.1.4        purrr_1.0.2        readr_2.1.5
## [17] tidyr_1.3.1        tibble_3.2.1       tidyverse_2.0.0    hrbrthemes_0.8.7
## [21] viridis_0.6.5      viridisLite_0.4.2  ggplot2_3.5.0
##
## loaded via a namespace (and not attached):
##  [1] nlme_3.1-164          fontquiver_0.2.1      httr_1.4.7
##  [4] tools_4.2.1           backports_1.4.1       utf8_1.2.4
##  [7] R6_2.5.1              lazyeval_0.2.2        colorspace_2.1-0
## [10] withr_3.0.0           tidyselect_1.2.1      mnormt_2.1.1
## [13] curl_5.2.1            compiler_4.2.1        extrafontdb_1.0
## [16] cli_3.6.2             plotly_4.10.4         fontBitstreamVera_0.1.1
## [19] labeling_0.4.3        scales_1.3.0          systemfonts_1.0.5
## [22] digest_0.6.35         rmarkdown_2.26        gfonts_0.2.0
## [25] pkgconfig_2.0.3       htmltools_0.5.7       extrafont_0.19
## [28] highr_0.10            fastmap_1.1.1         htmlwidgets_1.6.4
## [31] rlang_1.1.3           rstudioapi_0.15.0     httpcode_0.3.0
## [34] shiny_1.8.0           farver_2.1.1          generics_0.1.3
## [37] jsonlite_1.8.8        car_3.1-2             magrittr_2.0.3
## [40] Rcpp_1.0.12           munsell_0.5.0         fansi_1.0.6
## [43] abind_1.4-5           gdtools_0.3.5         lifecycle_1.0.4
## [46] stringi_1.8.3         yaml_2.3.8            carData_3.0-5
## [49] plyr_1.8.9            parallel_4.2.1        promises_1.2.1
## [52] crayon_1.5.2          lattice_0.22-5        hms_1.1.3
## [55] knitr_1.45            pillar_1.9.0          ggsignif_0.6.4
```

```
## [58] crul_1.4.0               glue_1.7.0              evaluate_0.23
## [61] fontLiberation_0.1.0     data.table_1.15.2       vctrs_0.6.5
## [64] tzdb_0.4.0               httpuv_1.6.14           Rttf2pt1_1.3.12
## [67] gtable_0.3.4             xfun_0.42               mime_0.12
## [70] xtable_1.8-4             broom_1.0.5             rstatix_0.7.2
## [73] later_1.3.2              timechange_0.3.0        ellipsis_0.3.2
```