mRNA stability assay after HNRNPC and UPF1 knockdown at 1h EGF stimulation

Luca Ducoli

01 July, 2024

This is the pipeline used to analyze the alternative splicing events which are regulated by HNRNPC and UPF1 during EGF stimulation.

1. PCA plot of mRNA stability data

First, we looked at PCA plots to evaluate the distribution of our samples.

```
#Needed libraries
library(tidyr)
library(dplyr)
library(tximportData)
library(GenomicFeatures)
library(tximport)
library(SummarizedExperiment)
library(tximeta)
library(fishpond)
library(org.Hs.eg.db)
library(ggfortify)
library(ggpubr)
library(ggrepel)
library(RColorBrewer)
library(pheatmap)
library(viridis)
library(broom)
library(PKNCA)
library(ggpubr)
library(limma)
library(DEP)
library(DESeq2)
library(EnsDb.Hsapiens.v86)
library(ggpmisc)
library(ggExtra)
library(MASS)
```

```
#Load salmon quant data for control with --numGibbsSamples 20
setwd("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/4_mRNA_stability_assay/
     4_DTE_timecourse/Analysis/0_PCA_eachTP")

#Sample names
sample_ctrl <- c("d0_ctrl1_quant", "d0_ctrl2_quant", "d1_ctrl1_quant", "d1_ctrl2_quant","d3_ctrl1
     _quant", "d3_ctrl2_quant","d4_ctrl1_quant", "d4_ctrl2_quant")
sample_HNRNPCKD <- c("d0_hn1_quant", "d0_hn2_quant", "d1_hn1_quant", "d1_hn2_quant","d3_hn1_quant
     ", "d3_hn2_quant", "d4_hn1_quant", "d4_hn2_quant")
sample_UPF1KD <- c("d0_up1_quant", "d0_up2_quant", "d1_up1_quant", "d1_up2_quant","d3_up1_quant",
      "d3_up2_quant", "d4_up1_quant", "d4_up2_quant")
```

```
 dir <- "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/4_mRNA_stability_assay
     /4_DTE_timecourse/Salmon_20/counts"

PCA_plot <- function(sample, type) {
  cat("Create colData\n")
  coldata <- data.frame(condition = factor(rep(c("D0", "D1", "D3", "D4"), each = 2)), repl =
    factor(rep(c("1","2"),4)))
  coldata$names <- sample
  coldata$files <- file.path(dir, coldata$names, "quant.sf")
  all(file.exists(coldata$files))

  cat("Create Summarized exp\n")
  se <- tximeta(coldata, skipMeta = TRUE, skipSeqinfo = TRUE)
  se <- scaleInfReps(se)
  se <- labelKeep(se)
  se <- se[mcols(se)$keep,]
  scaled_counts <- (se@assays@data$infRep1+se@assays@data$infRep2+se@assays@data$infRep3+
    se@assays@data$infRep4+se@assays@data$infRep5
                       +se@assays@data$infRep6+se@assays@data$infRep7+se@assays@data$infRep8+
    se@assays@data$infRep9+se@assays@data$infRep10
                       +se@assays@data$infRep11+se@assays@data$infRep12+se@assays@data$infRep13+
    se@assays@data$infRep14+se@assays@data$infRep15
                       +se@assays@data$infRep16+se@assays@data$infRep17+se@assays@data$infRep18+
    se@assays@data$infRep19+se@assays@data$infRep20)/20
  write.table(scaled_counts, file = paste("Count_table_", type, ".txt", sep = ""), quote = FALSE,
     sep = "\t", row.names = TRUE)

  cat("Generate PCA\n")
  pca_res <- prcomp(t(scaled_counts))

  psi.pca.pc <- data.frame(pca_res$x, sample = colnames(scaled_counts)) %>%
    mutate(., condition = paste(rep(type, 8), c('D0 R1', 'D0 R2', "D1 R1", "D1 R2", "D3 R1", "D3
    R2","D4 R1", "D4 R2"), sep = "-"))

  psi.pca.summary <- summary(pca_res)$importance
  pc1var = round(psi.pca.summary[2,1] * 100, 1)
  pc2var = round(psi.pca.summary[2,2] * 100, 1)

  colors <- brewer.pal(8, 'Set1')

  PCA <- ggplot(psi.pca.pc, aes(x = PC1, y = PC2, color = condition)) + geom_point(size = 5)  +
    scale_color_manual(values = colors) + theme_linedraw() + xlab(paste('PC1, ', pc1var, '%
    explained var.')) +
    ylab(paste('PC2, ', pc2var, '% explained var.')) + theme(legend.position = "none") + geom_text
    _repel(aes(label = condition), box.padding = 0.5)

  return(list(se = se, PCA = PCA, scaled_counts = scaled_counts ))
}

 Control <- PCA_plot(sample_ctrl, "Ctrl")
```

```
## Create colData
## Create Summarized exp
## Generate PCA
```

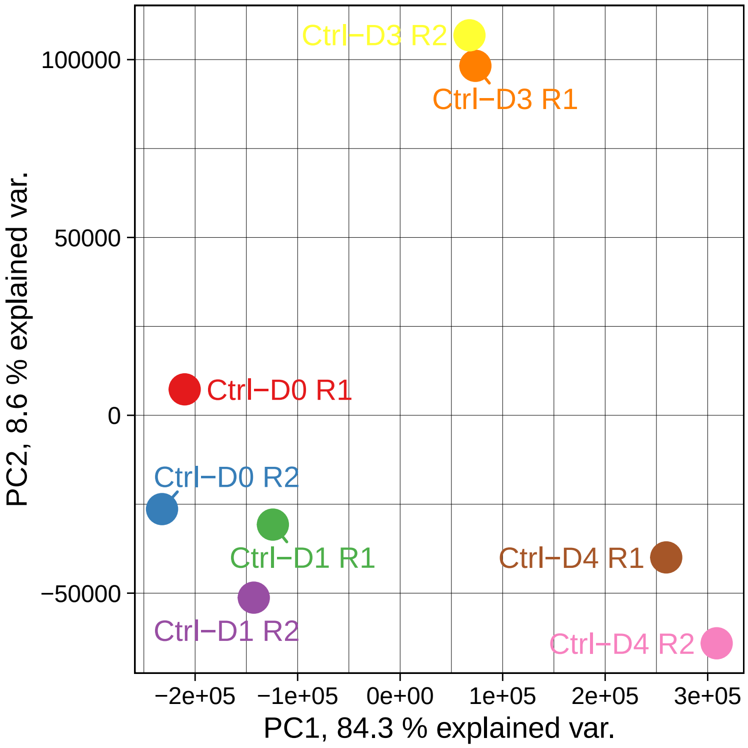
```
HNRNPCKD <- PCA_plot(sample_HNRNPCKD, "HNRNPC-KD")
```

```
## Create colData
## Create Summarized exp
## Generate PCA
```

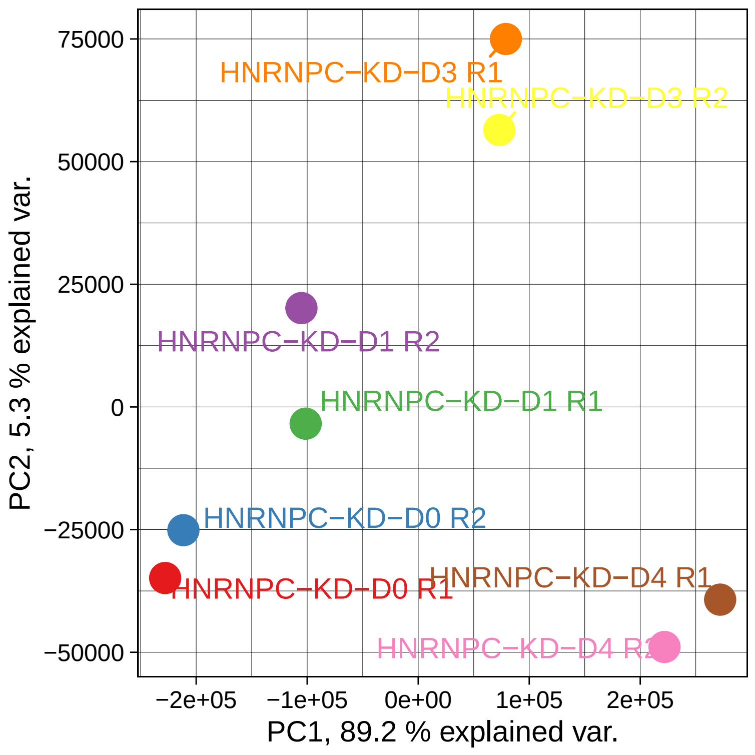
```
UPF1KD <- PCA_plot(sample_UPF1KD, "UPF1-KD")
```

```
## Create colData
## Create Summarized exp
## Generate PCA
```

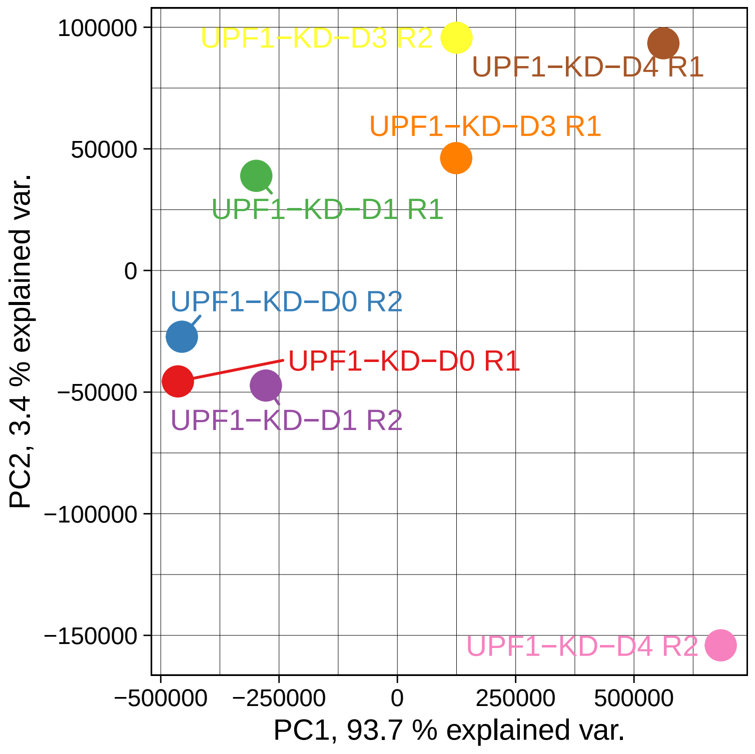
```
Control$PCA
```



```
HNRNPCKD$PCA
```



```
UPF1KD$PCA
```



```
#Save the plot as pdf
```

```
pdf("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/4_mRNA_stability_assay/4_
    DTE_timecourse/Analysis/0_PCA_eachTP/PCA_mRNAstab.pdf")
Control$PCA
HNRNPCKD$PCA
UPF1KD$PCA
dev.off()
```

2. Differential stability (DS) analysis against control samples

This analysis is to determine the transcripts that shows disregulated stability compared to control siRNAs.

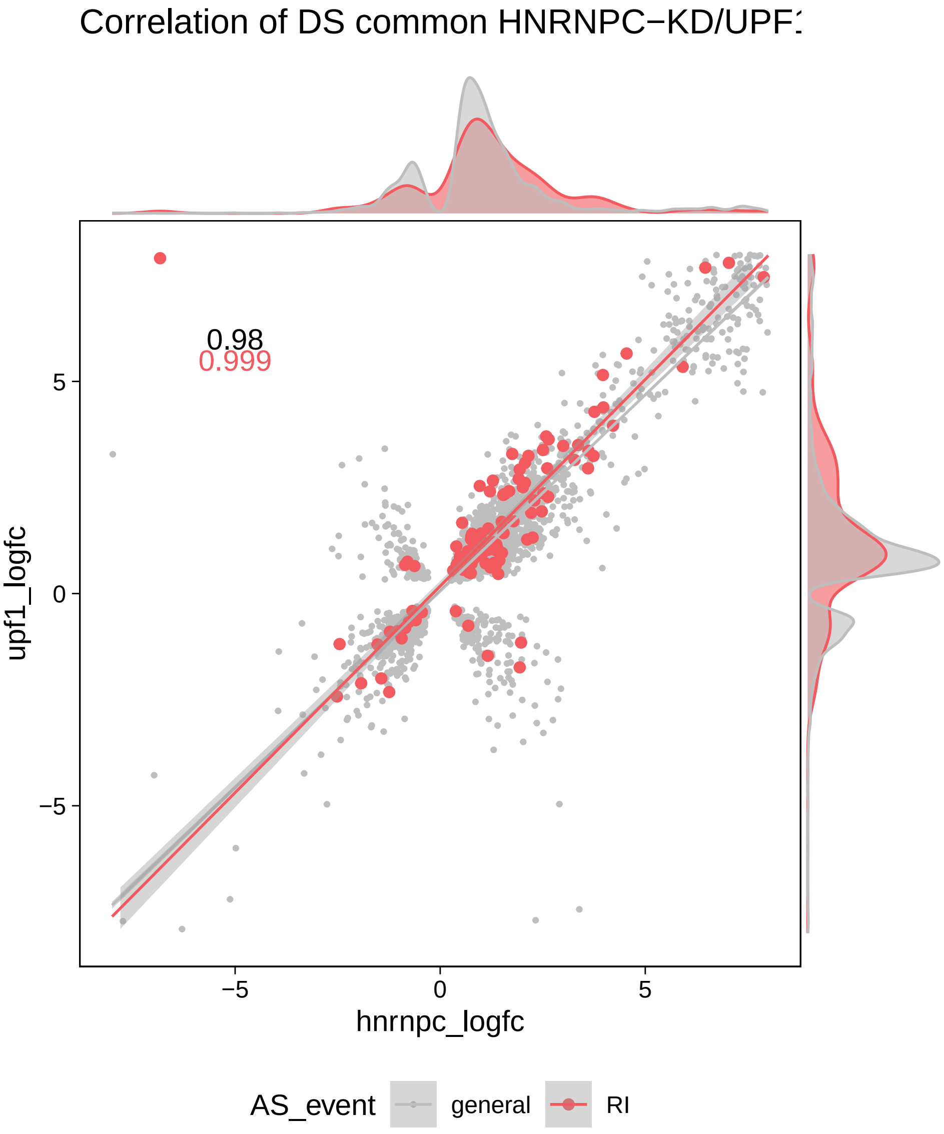
```
#Load annotation
txdb <- loadDb("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/8_DTE_analysis
    /0_UPF1_HNRNPC/1_DTE/Genome/gencode.v39.annotation.sqlite")
k <- keys(txdb, keytype = "TXNAME")
tx2gene <- AnnotationDbi::select(txdb, k, "GENEID", "TXNAME")
tx2gene$geneID <- gsub("\\..*", "", tx2gene$GENEID)
geneIDs1 <- ensembldb::select(EnsDb.Hsapiens.v86, keys= tx2gene$geneID, keytype = "GENEID",
    columns = c("SYMBOL","GENEID"))
colnames(geneIDs1)[2] <- "geneID"
tx2gene <- merge(tx2gene, geneIDs1, by = "geneID", all.x = TRUE)
write.table(tx2gene, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/4
    _mRNA_stability_assay/4_DTE_timecourse/Analysis/1_DTE/Annotation_transcripts_genes.txt",
    quote = FALSE, sep = "\t", row.names = FALSE)
```

Regression analysis of DS RI transcripts

```
#Load the expression table
control.expr <- as.data.frame(Control$scaled_counts)
control.expr$TXid <- rownames(control.expr)
rownames(control.expr) <- NULL
control.expr <- control.expr[,c(9,1:8)]

hnrnpc.expr <- as.data.frame(HNRNPCKD$scaled_counts)
hnrnpc.expr$TXid <- rownames(hnrnpc.expr)
rownames(hnrnpc.expr) <- NULL

upf1.expr <- as.data.frame(UPF1KD$scaled_counts)
upf1.expr$TXid <- rownames(upf1.expr)
rownames(upf1.expr) <- NULL

#Merge scaled
mrna.stap.expr <- merge(control.expr,hnrnpc.expr, by="TXid")
mrna.stap.expr <- merge(mrna.stap.expr,upf1.expr, by="TXid")

DTE.high.expr2 <- mrna.stap.expr

#Limma
design <- read.delim("/Users/lducoli/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2
    _EGF/4_mRNA_stability_assay/4_DTE_timecourse/Analysis/1_DTE/0_design.txt")

colnames(DTE.high.expr2)[1] <- "name"
DTE.high.expr2$ID <- 1:length(rownames(DTE.high.expr2))
columns <- c(grep("d\\d+", colnames(DTE.high.expr2)))
mrnastab_se <- make_se(DTE.high.expr2, columns, design)

model <- model.matrix(~ time + time:rbp, colData(mrnastab_se))
mrnastab_fit1_int = lmFit(assay(mrnastab_se), design = model)
```

```
mrnastab_fit2_int <- eBayes(mrnastab_fit1_int)
mrnastab_int_res <- topTable(mrnastab_fit2_int, coef = c("timeT1:rbphnrnpc", "timeT3:rbphnrnpc",
    "timeT4:rbphnrnpc"), number = length(rownames(mrnastab_se)))
mrnastab_int_res2 <- topTable(mrnastab_fit2_int, coef = c("timeT1:rbpupf1", "timeT3:rbpupf1", "
    timeT4:rbpupf1"), number = length(rownames(mrnastab_se)))
write.table(mrnastab_int_res, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_
    SG/2_EGF/4_mRNA_stability_assay/4_DTE_timecourse/Analysis/1_DTE/HNRNPC_Limma_results.txt",
    row.names = TRUE, quote = FALSE, sep = "\t")
write.table(mrnastab_int_res2, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_
    SG/2_EGF/4_mRNA_stability_assay/4_DTE_timecourse/Analysis/1_DTE/UPF1_Limma_results.txt", row.
    names = TRUE, quote = FALSE, sep = "\t")

#Max FC
mrnastab_int_res$max_lfc <- pmax(mrnastab_int_res$timeT1.rbphnrnpc,mrnastab_int_res$timeT3.
    rbphnrnpc, mrnastab_int_res$timeT4.rbphnrnpc)
mrnastab_int_res$min_lfc <- pmin(mrnastab_int_res$timeT1.rbphnrnpc,mrnastab_int_res$timeT3.
    rbphnrnpc, mrnastab_int_res$timeT4.rbphnrnpc)
mrnastab_int_res$max_lfc_all <- ifelse(abs(mrnastab_int_res$max_lfc) > abs(mrnastab_int_res$min_
    lfc),  mrnastab_int_res$max_lfc, mrnastab_int_res$min_lfc)
mrnastab_int_res$TXNAME <- rownames(mrnastab_int_res)

mrnastab_int_res2$max_lfc <- pmax(mrnastab_int_res2$timeT1.rbpupf1,mrnastab_int_res2$timeT3.
    rbpupf1, mrnastab_int_res2$timeT4.rbpupf1)
mrnastab_int_res2$min_lfc <- pmin(mrnastab_int_res2$timeT1.rbpupf1,mrnastab_int_res2$timeT3.
    rbpupf1, mrnastab_int_res2$timeT4.rbpupf1)
mrnastab_int_res2$max_lfc_all <- ifelse(abs(mrnastab_int_res2$max_lfc) > abs(mrnastab_int_res2$
    min_lfc),  mrnastab_int_res2$max_lfc, mrnastab_int_res2$min_lfc)
mrnastab_int_res2$TXNAME <- rownames(mrnastab_int_res2)

#Prepare DTE results
logfc.hnrnpc <- mrnastab_int_res %>% dplyr::select(TXNAME, max_lfc_all, adj.P.Val)
logfc.upf1 <- mrnastab_int_res2 %>% dplyr::select(TXNAME, max_lfc_all, adj.P.Val)

colnames(logfc.hnrnpc) <- c("TX_id", "hnrnpc_logfc", "hnrnpc_adjpval")
colnames(logfc.upf1) <- c("TX_id", "upf1_logfc", "upf1_adjpval")

logfc.table.vsupf1 <- merge(logfc.hnrnpc, logfc.upf1, by = "TX_id")
logfc.table.vsupf1.sign <- subset(logfc.table.vsupf1, hnrnpc_adjpval < 0.05 & upf1_adjpval <
    0.05)

#Merge logfc table with rMATs transcript
DTE.high <- read.delim("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/8_DTE_
    analysis/0_UPF1_HNRNPC/1_DTE/AS_HNRNPC_UPF1_transcripts.txt", header = TRUE)

as.tr.fc <- merge(logfc.table.vsupf1.sign, unique(DTE.high %>% dplyr::select(TX_id, TX_geneID, AS
    _event)), by = "TX_id")
write.table(as.tr.fc, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/
    4_mRNA_stability_assay/4_DTE_timecourse/Analysis/1_DTE/Limma_sign_transcript_upset.txt", row.
    names = TRUE, quote = FALSE, sep = "\t")

#Regression analysis
all <- setdiff(logfc.table.vsupf1.sign$TX_id, as.tr.fc$TX_id)
logfc.table.vsupf1.sign2 <- subset(logfc.table.vsupf1.sign, TX_id %in% all)
logfc.table.vsupf1.sign2$AS_event <- "general"
logfc.table.vsupf1.sign2$type <- "general"

as.tr.fc.plot <- as.tr.fc
as.tr.fc.plot$type <- "AS_event"

model <- rlm(upf1_logfc ~ hnrnpc_logfc, data = logfc.table.vsupf1.sign2)
model2 <- rlm(upf1_logfc ~ hnrnpc_logfc, data = as.tr.fc.plot)

rlm_rsquare <- function(model) {
# Extract the weights and residuals
weights <- model$w
residuals <- model$resid
fitted_values <- model$fitted.values
response <- logfc.table.vsupf1.sign2$upf1_logfc
# Calculate the weighted sum of squares
weighted_ss_residuals <- sum(weights * residuals^2)
weighted_ss_total <- sum(weights * (response - mean(response))^2)

# Calculate the weighted R-squared
weighted_r_squared <- 1 - (weighted_ss_residuals / weighted_ss_total)

return(weighted_r_squared)
}
model.rsquare <- rlm_rsquare(model)
model2.rsquare <- rlm_rsquare(model2)

chow.test <- gap::chow.test(y1=logfc.table.vsupf1.sign2$upf1_logfc,x1=logfc.table.vsupf1.sign2$
    hnrnpc_logfc,y2=as.tr.fc.plot$upf1_logfc,x2=as.tr.fc.plot$hnrnpc_logfc)
model.res <- data.frame(general = model.rsquare, RI_event = model2.rsquare, chow.pvalue = chow.
    test[[4]])
write.table(model.res, "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/4_mRNA
    _stability_assay/4_DTE_timecourse/Analysis/1_DTE/Chowtest_allASevents.txt", quote = F, sep =
    "\t")

as.tr.fc.plot <- rbind(logfc.table.vsupf1.sign2, as.tr.fc.plot %>% dplyr::select(-TX_geneID))
as.tr.fc.plot$AS_event <- factor(as.tr.fc.plot$AS_event)

#Generate the scatter plot
ggplot3 <- ggplot(data=as.tr.fc.plot, aes(x=hnrnpc_logfc, y=upf1_logfc, group=type)) +
  geom_point(aes(color = AS_event, size = AS_event)) +
  scale_size_manual(values=c(0.5,1.5))+
  geom_smooth(aes(group=AS_event, color=AS_event), method='rlm', formula= y~x, size=0.5,
    fullrange=TRUE) +
  scale_color_manual(values=c(general = "grey", RI = "#f25a5f"))+
  annotate(geom="text", x=-5, y=6, label=round(model.rsquare,2), color="black") +
  annotate(geom="text", x=-5, y=5.5, label=round(model2.rsquare,3), color="#f25a5f") +
  xlim(-8, 8) +
  ylim(-8, 8) +
  theme_linedraw() + theme(panel.grid.major = element_blank(), legend.position = "bottom",
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        axis.line = element_blank()) + ggtitle("Correlation of DS common HNRNPC-KD/UPF1-KD AS
    transcripts")
ggplot3 <- ggMarginal(ggplot3, groupColour = TRUE, groupFill = TRUE, type="density", alpha=0.6,
    size = 5)
```

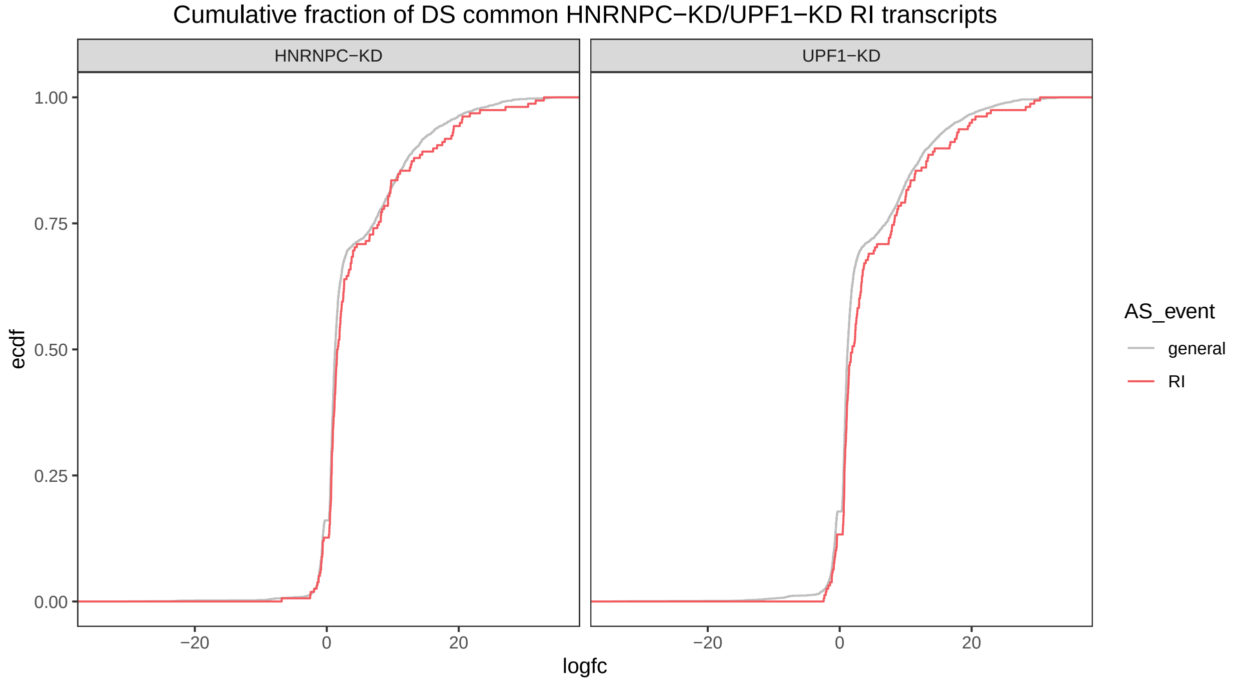
```
ggplot3
```



```
#Save the plot as pdf
pdf("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/4_mRNA_stability_assay/4_
    DTE_timecourse/Analysis/1_DTE/Regression_DST_HNRNPC_UPF1.pdf", height = 6, width = 5)
ggplot3
dev.off()
```

Cumulative fraction of DS RI transcripts

```
#Cumulative fraction of UPF1 RI
as.tr.fc.cdf <- as.tr.fc.plot %>% dplyr::select(TX_id, hnrnpc_logfc, AS_event) %>% dplyr::rename(
     logfc = hnrnpc_logfc)
as.tr.fc.cdf$rbp <- "HNRNPC-KD"

as.tr.fc.cdf2 <- as.tr.fc.plot %>% dplyr::select(TX_id, upf1_logfc, AS_event) %>% dplyr::rename(
     logfc = upf1_logfc)
as.tr.fc.cdf2$rbp <- "UPF1-KD"
as.tr.fc.cdf <- rbind(as.tr.fc.cdf, as.tr.fc.cdf2)

cdf <- ggplot(data=as.tr.fc.cdf, aes(x=logfc, group=AS_event, colour=AS_event)) +
  stat_ecdf() +
  ggtitle("Cumulative fraction of DS common HNRNPC-KD/UPF1-KD RI transcripts") +
  scale_color_manual(values = c(A3SS = "#7959b7", A5SS = "#00b7e6", general = "grey", MXE = "#
    ff955e", RI = "#f25a5f", SE = "#4487ab")) +
  theme_bw() +
  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)) + facet_grid(cols=vars(rbp))

cdf
```



```
#Save the plot as pdf
pdf("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/4_mRNA_stability_assay/4_
    DTE_timecourse/Analysis/1_DTE/ECDF_DST_HNRNPC_UPF1.pdf", height = 5, width = 9)
cdf
dev.off()
```

```
#KS test
hnrncp.kstest <- data.frame( ks.pvalue =  c(ks.test(as.tr.fc.cdf$logfc[as.tr.fc.cdf$AS_event == "
    general" & as.tr.fc.cdf$rbp == "HNRNPC-KD"], as.tr.fc.cdf$logfc[as.tr.fc.cdf$AS_event == "RI"
     & as.tr.fc.cdf$rbp == "HNRNPC-KD"] , alternative = 'greater')[["p.value"]]))

UPF1.kstest <- data.frame( ks.pvalue =  c(ks.test(as.tr.fc.cdf$logfc[as.tr.fc.cdf$AS_event == "
    general" & as.tr.fc.cdf$rbp == "UPF1-KD"], as.tr.fc.cdf$logfc[as.tr.fc.cdf$AS_event == "RI" &
     as.tr.fc.cdf$rbp == "UPF1-KD"] , alternative = 'greater')[["p.value"]]))

hnrncp.kstest
```

```
##    ks.pvalue
## 1 0.03995025
```

```
UPF1.kstest
```

```
##     ks.pvalue
## 1 0.001170439
```

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] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] MASS_7.3-60.0.1             ggExtra_0.10.1
##  [3] ggpmisc_0.5.5               ggpp_0.5.6
##  [5] EnsDb.Hsapiens.v86_2.99.0   ensembldb_2.20.2
##  [7] AnnotationFilter_1.22.0     DESeq2_1.38.3
##  [9] DEP_1.18.0                  limma_3.54.2
## [11] PKNCA_0.10.2                broom_1.0.5
## [13] viridis_0.6.5               viridisLite_0.4.2
## [15] pheatmap_1.0.12             RColorBrewer_1.1-3
## [17] ggrepel_0.9.5               ggpubr_0.6.0
## [19] ggfortify_0.4.16            ggplot2_3.5.0
## [21] org.Hs.eg.db_3.15.0         fishpond_2.2.0
## [23] tximeta_1.17.2              SummarizedExperiment_1.28.0
## [25] MatrixGenerics_1.10.0       matrixStats_1.2.0
## [27] tximport_1.24.0             GenomicFeatures_1.48.4
## [29] AnnotationDbi_1.60.2        Biobase_2.58.0
## [31] GenomicRanges_1.50.2        GenomeInfoDb_1.34.9
## [33] IRanges_2.32.0              S4Vectors_0.36.2
## [35] BiocGenerics_0.44.0         tximportData_1.24.0
## [37] dplyr_1.1.4                 tidyr_1.3.1
##
## loaded via a namespace (and not attached):
##   [1] shinydashboard_0.7.2          utf8_1.2.4
##   [3] gmm_1.8                       tidyselect_1.2.1
##   [5] htmlwidgets_1.6.4             RSQLite_2.3.5
##   [7] grid_4.2.1                    BiocParallel_1.32.6
##   [9] norm_1.0-11.1                 munsell_0.5.0
##  [11] codetools_0.2-19              preprocessCore_1.60.2
##  [13] DT_0.32                       miniUI_0.1.1.1
##  [15] withr_3.0.0                   colorspace_2.1-0
##  [17] filelock_1.0.3              highr_0.10
##  [19] knitr_1.45                  rstudioapi_0.15.0
##  [21] SingleCellExperiment_1.20.1 ggsignif_0.6.4
##  [23] mzID_1.36.0                 Rdpack_2.6
##  [25] labeling_0.4.3              GenomeInfoDbData_1.2.9
##  [27] farver_2.1.1                bit64_4.0.5
##  [29] gap.datasets_0.0.6          vctrs_0.6.5
##  [31] generics_0.1.3              xfun_0.42
##  [33] BiocFileCache_2.4.0         R6_2.5.1
##  [35] doParallel_1.0.17           clue_0.3-65
##  [37] locfit_1.5-9.9              MsCoreUtils_1.10.0
##  [39] bitops_1.0-7                cachem_1.0.8
##  [41] DelayedArray_0.24.0         assertthat_0.2.1
##  [43] vroom_1.6.5                 promises_1.2.1
##  [45] BiocIO_1.6.0                scales_1.3.0
##  [47] gtable_0.3.4                affy_1.76.0
##  [49] sandwich_3.1-0              MatrixModels_0.5-3
##  [51] rlang_1.1.3                 mzR_2.32.0
##  [53] splines_4.2.1               GlobalOptions_0.1.2
##  [55] rtracklayer_1.56.1          rstatix_0.7.2
##  [57] lazyeval_0.2.2              impute_1.72.3
##  [59] BiocManager_1.30.22         yaml_2.3.8
##  [61] abind_1.4-5                 backports_1.4.1
##  [63] httpuv_1.6.14               tools_4.2.1
##  [65] affyio_1.68.0               ellipsis_0.3.2
##  [67] polynom_1.4-1               MSnbase_2.24.2
##  [69] Rcpp_1.0.12                 plyr_1.8.9
##  [71] progress_1.2.3              zlibbioc_1.44.0
##  [73] purrr_1.0.2                 RCurl_1.98-1.14
##  [75] prettyunits_1.2.0           GetoptLong_1.0.5
##  [77] zoo_1.8-12                  cluster_2.1.6
##  [79] svMisc_1.2.3                magrittr_2.0.3
##  [81] SparseM_1.81                circlize_0.4.16
##  [83] pcaMethods_1.90.0           mvtnorm_1.2-4
##  [85] ProtGenerics_1.30.0         hms_1.1.3
##  [87] mime_0.12                   evaluate_0.23
##  [89] xtable_1.8-4                XML_3.99-0.16.1
##  [91] gridExtra_2.3               shape_1.4.6.1
##  [93] compiler_4.2.1              biomaRt_2.52.0
##  [95] tibble_3.2.1                ncdf4_1.22
##  [97] crayon_1.5.2                htmltools_0.5.7
##  [99] mgcv_1.9-1                  tzdb_0.4.0
## [101] later_1.3.2                 geneplotter_1.76.0
## [103] DBI_1.2.2                   dbplyr_2.4.0
## [105] ComplexHeatmap_2.14.0       tmvtnorm_1.6
## [107] rappdirs_0.3.3              readr_2.1.5
## [109] Matrix_1.6-5                car_3.1-2
## [111] cli_3.6.2                   vsn_3.66.0
## [113] imputeLCMD_2.1              rbibutils_2.2.16
## [115] parallel_4.2.1              pkgconfig_2.0.3
## [117] GenomicAlignments_1.34.1    MALDIquant_1.22.2
## [119] xml2_1.3.6                  foreach_1.5.2
## [121] annotate_1.76.0             XVector_0.38.0
## [123] stringr_1.5.1               digest_0.6.35
## [125] Biostrings_2.66.0           rmarkdown_2.26
## [127] gap_1.5-3                   restfulr_0.0.15
## [129] curl_5.2.1                  quantreg_5.97
## [131] shiny_1.8.0                 Rsamtools_2.14.0
## [133] gtools_3.9.5                rjson_0.2.21
## [135] lifecycle_1.0.4             nlme_3.1-164
## [137] jsonlite_1.8.8              carData_3.0-5
## [139] fansi_1.0.6                 pillar_1.9.0
## [141] lattice_0.22-5              survival_3.5-8
## [143] KEGGREST_1.38.0             fastmap_1.1.1
## [145] httr_1.4.7                  interactiveDisplayBase_1.34.0
## [147] glue_1.7.0                  png_0.1-8
## [149] iterators_1.0.14            BiocVersion_3.15.2
## [151] bit_4.0.5                   stringi_1.8.3
## [153] blob_1.2.4                  AnnotationHub_3.4.0
## [155] memoise_2.0.1
```