RNA-seq analysis of time-course EGF data

Luca Ducoli

01 July, 2024

This is the pipeline used to analyze the EGF time-course RNAseq data. We collected the following time points: 0min, 15min, 30min, 60min. This experiment was done in A431 cells.

1. Load the counts from Salmon

```
#Needed libraries
library(DESeq2)
library(pheatmap)
library(ggplot2)
library(MASS)
library(ggpubr)
library(ggExtra)
library(forcats)
library(tximportData)
library(GenomicFeatures)
library(tximport)
library(gprofiler2)
library(viridis)
library(AnnotationDbi)
library(org.Hs.eg.db)
library(eulerr)
library(SuperExactTest)
library(tidyverse)
```

```
#Prepare sample list
dir <- "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/9_EGF_TC/1_Salmon_
    quant/counts"

#Prepare a sample list file in Excel and store in the counts folder
samples <- read.table("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/9_EGF_
    TC/1_Salmon_quant/sample_metadata.txt", header = TRUE)
samples
```

```
##       sample              run
## 1   EGF_T0_1   EGF_T0_1_quant
## 2   EGF_T0_2   EGF_T0_2_quant
## 3  EGF_T15_1  EGF_T15_1_quant
## 4  EGF_T15_2  EGF_T15_2_quant
## 5  EGF_T30_1  EGF_T30_1_quant
## 6  EGF_T30_2  EGF_T30_2_quant
## 7  EGF_T60_1  EGF_T60_1_quant
## 8  EGF_T60_2  EGF_T60_2_quant
```

```
files <- file.path(dir, samples$run, "quant.sf")
names(files) <- paste0(samples$sample)

#Load annotation from irCLIP-RNP project
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")

#Import data and save counts and txi object
txi.salmon <- tximport(files, type = "salmon", tx2gene = tx2gene)
head(txi.salmon$counts)
```

```
##                     EGF_T0_1 EGF_T0_2 EGF_T15_1 EGF_T15_2 EGF_T30_1 EGF_T30_2
## ENSG00000000003.15  282.000       71       109        36        83       105
## ENSG00000000005.6     0.000        0         0         0         0         0
## ENSG00000000419.14  153.999       15        11         4         7        20
## ENSG00000000457.14   14.000        4         6         4         5         4
## ENSG00000000460.17   37.000        9        12         3         1        14
## ENSG00000000938.13    0.000        0         0         0         3         0
##                     EGF_T60_1 EGF_T60_2
## ENSG00000000003.15          8        16
## ENSG00000000005.6           0         0
## ENSG00000000419.14          6         1
## ENSG00000000457.14          0         0
## ENSG00000000460.17          0         1
## ENSG00000000938.13          1         0
```

```
write.table(txi.salmon$counts, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_
    SG/2_EGF/9_EGF_TC/1_Salmon_quant/RNA-Seq_egftc_salmon_counts_all.txt", row.names = TRUE,
    quote = FALSE, sep = "\t")
saveRDS(txi.salmon, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/9_
    EGF_TC/1_Salmon_quant/RNA-Seq_egftc_txi_object.rds")
```

2. Differential expression analysis

Here, we performed differential expression analysis at the gene-level using DESeq2.

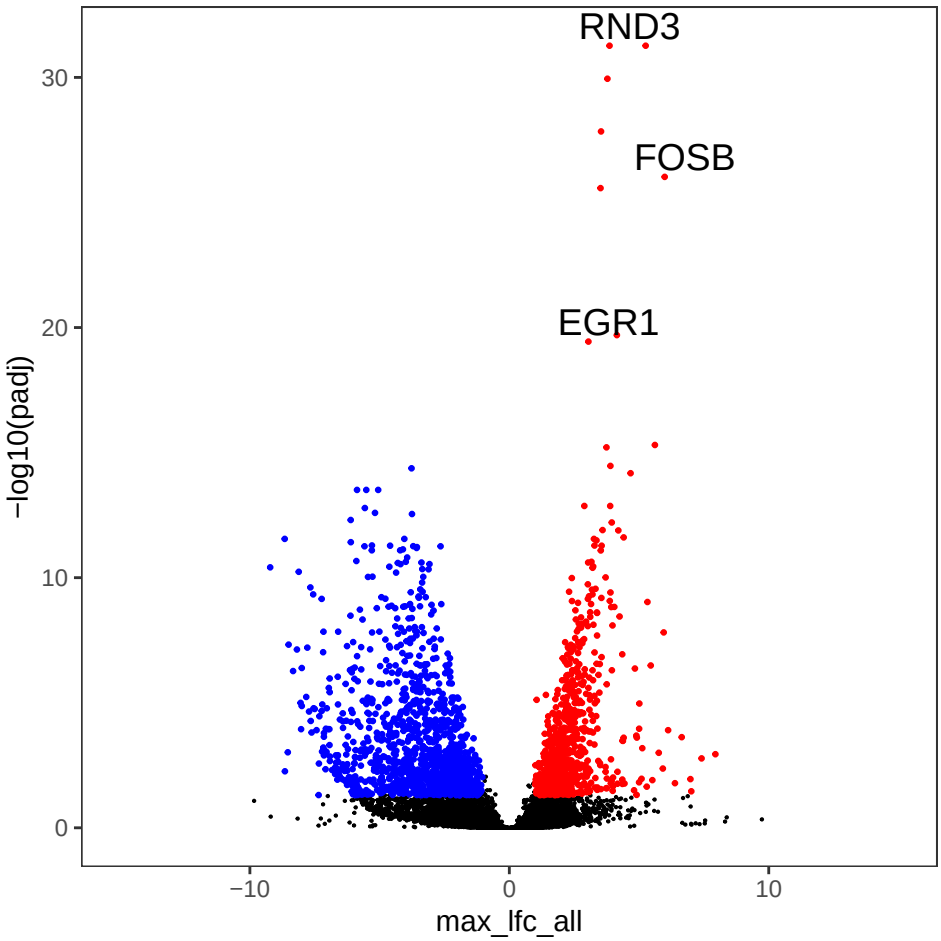
```
#Get sample colData
sampleTable <- read.table("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/9_
    EGF_TC/1_Salmon_quant/coldata.txt", header = TRUE)
rownames(sampleTable) <- colnames(txi.salmon$counts)
```

```
#Generate object
dds <- DESeqDataSetFromTximport(txi.salmon, sampleTable, ~ condition)
keep <- rowSums(counts(dds)) >= 1
dds <- dds[keep,]

#Generate normalized counts
dds <- estimateSizeFactors(dds)
normalized_counts <- counts(dds, normalized=TRUE)
write.table(normalized_counts, file="~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG
    /2_EGF/9_EGF_TC/2_Genelevel_analysis/RNA-Seq_egftc_norm_count_table.txt", sep="\t", quote=F,
    col.names=NA)
```

```
#Run DESeq2 differential expression analysis
dds <- DESeq(dds, test="LRT", reduced = ~ 1)
res <- as.data.frame(results(dds))
res15 <- as.data.frame(results(dds, name="condition_T15_vs_T0"))
res30 <- as.data.frame(results(dds, name="condition_T30_vs_T0"))
res60 <- as.data.frame(results(dds, name="condition_T60_vs_T0"))

res <- cbind(res, logFC_T15 = res15$log2FoldChange, logFC_T30 = res30$log2FoldChange, logFC_T60 =
      res60$log2FoldChange)
res$max_lfc <- pmax(res$logFC_T15,res$logFC_T30, res$logFC_T60)
res$min_lfc <- pmin(res$logFC_T15,res$logFC_T30, res$logFC_T60)
res$max_lfc_all <- ifelse(abs(res$max_lfc) > abs(res$min_lfc), res$max_lfc, res$min_lfc)

#Write significant results
res$geneIDv <- rownames(res)
res$geneID <- gsub("\\..*","",rownames(res))
res$external_gene_name = mapIds(org.Hs.eg.db, keys=res$geneID, column="SYMBOL", keytype="ENSEMBL"
     , multiVals="first")
write.table(res, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/9_EGF
    _TC/2_Genelevel_analysis/RNA-Seq_egtc_DESeq2.txt", quote = FALSE, row.names = F, sep = "\t")

res.sign <- subset(res, padj < 0.05 & abs(max_lfc_all) > 1)
write.table(res.sign, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/
    9_EGF_TC/2_Genelevel_analysis/RNA-Seq_egtc_DESeq2_sign.txt", quote = FALSE, row.names = F,
    sep = "\t")

res.sign.up <- subset(res, padj < 0.05 & max_lfc_all > 1)
res.sign.dwn <- subset(res, padj < 0.05 & max_lfc_all < -1)
```

Volcano plot of significant genes

```
#Volcano plot
ggplot(data=res, aes(x=max_lfc_all, y=-log10(padj))) +
  geom_point(size = 0.1) +
  geom_point(data = res.sign.up, aes(x=max_lfc_all, y = -log10(padj)), color = "red", size = 0.5)
     +
  geom_point(data = res.sign.dwn, aes(x=max_lfc_all, y = -log10(padj)), color = "blue", size =
    0.5) +
  ggrepel::geom_text_repel(data = dplyr::filter(res.sign.up, external_gene_name %in% c("FOSB", "
    EGR1", "RND3")), aes(label = external_gene_name), size = 5, box.padding = unit(0.1, "lines"),
      point.padding = unit(0.1, "lines"), segment.size = 0.5) +
  xlim(-15, 15) +
  # ylim(-2, 5) +
  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)) +
  theme(legend.position="none")
```




```
#Save the plot as pdf
pdf("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/9_EGF_TC/2_Genelevel_
    analysis/RNA-Seq_egftc_Volcano_plot.pdf", height = 5, width = 5)
ggplot(data=res, aes(x=log2FoldChange, y=-log10(padj))) +
  geom_point(size = 0.1) +
  geom_point(data = res.sign.up, aes(x=log2FoldChange, y = -log10(padj)), color = "red", size =
    0.5) +
  geom_point(data = res.sign.dwn, aes(x=log2FoldChange, y = -log10(padj)), color = "blue", size =
    0.5) +
  ggrepel::geom_text_repel(data = dplyr::filter(res.sign.up, external_gene_name %in% c("FOSB", "
    EGR1", "RND3")), aes(label = external_gene_name), size = 5, box.padding = unit(0.1, "lines"),
     point.padding = unit(0.1, "lines"), segment.size = 0.5) +
  xlim(-15, 15) +
  # ylim(-2, 5) +
  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)) +
  theme(legend.position="none")
dev.off()
```

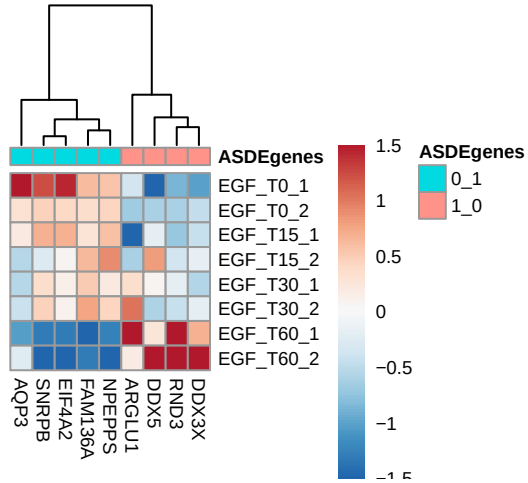
Heatmap with RI events that are co-bound and co-regulated by HNRNPC and UPF1.

```
#Heatmap with AS events
```

```
geneAS <- read.delim("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/8_DTE_
    analysis/0_UPF1_HNRNPC/0_Overlap/HNRNPC_UPF1_RIevents.txt", header = TRUE)
geneAS$geneIDv <- sapply(strsplit(geneAS$region, "_"), function(x) x[1])
geneAS$geneName <- sapply(strsplit(geneAS$region, "_"), function(x) x[2])

geneAScob <- read.delim("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/Seq/4_EGF_tc/4_
    Visualization/2_Heatmap/HNRNPC_UPF1_cobound_genes.txt", header = TRUE)
geneAS <- subset(geneAS, geneName %in% geneAScob$region)

# Load files
s4 <- list(EGF_TC = res.sign$geneIDv,
           Splicing = unique(geneAS$geneIDv))

# #Hypergeometric test
n = length(unique(c(rownames(assay(dds)))))
results <- supertest(x=s4, n=n, degree=c(1:2))

# #Print P-value of interaction
test_results <- summary(results)$Table
write.table(test_results, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_
    EGF/9_EGF_TC/2_Genelevel_analysis/test_results_overlap_splicing_egf.txt", row.names = FALSE,
    sep = "\t", quote = F)

fromList <- 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)
}

#Up and down genes separately
s5 <- list(EGF_TC_up = res.sign.up$geneIDv,
           EGF_TC_dwn = res.sign.dwn$geneIDv,
           Splicing = unique(geneAS$geneIDv))

#Binary table with colnames:
sign.AS.egf <- fromList(s5)
sign.AS.egf$geneID <- gsub("\\..*","",rownames(sign.AS.egf))
write.table(sign.AS.egf, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_
    EGF/9_EGF_TC/2_Genelevel_analysis/Overlap_splicing_egftc.txt", row.names = TRUE, sep = "\t",
    quote = F)
```

```
#Prepare the matrix for heatmap
res.sign.egftc <- subset(res.sign, geneIDv %in%  unique(geneAS$geneIDv))
write.table(res.sign.egftc, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/
    2_EGF/9_EGF_TC/2_Genelevel_analysis/RNA-Seq_egtc_DESeq2_sign_RI.txt", quote = FALSE, row.
    names = F, sep = "\t")
mat_hm <- normalized_counts
mat_hm <- merge(mat_hm, res.sign.egftc[,12:15], by = "row.names", sort = F)
rownames(mat_hm) <- mat_hm$external_gene_name
mat_hm <- as.matrix(mat_hm[,grep("EGF", colnames(mat_hm))])
mat_hm <- log2(mat_hm+min(mat_hm[mat_hm > 0]))
mat_hm_scaled <- scale(t(mat_hm), scale = TRUE, center = TRUE)

#Annotation
sign.AS.egf.sub <- merge(sign.AS.egf, res.sign.egftc[,12:15], by = "row.names", sort = F)
annotation_col <- data.frame(ASDEgenes = paste(sign.AS.egf.sub$EGF_TC_up, sign.AS.egf.sub$EGF_TC_
    dwn, sep = "_"))
rownames(annotation_col) <- sign.AS.egf.sub$external_gene_name
```

```
my.breaks <- c(seq(-1.5, -0.025, by=0.025), seq(0.025, 1.5, 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))

#Generate the heatmap
pheatmap(mat_hm_scaled,
  main=NA,
  color = my.colors,
  breaks = my.breaks,
  annotation_col = annotation_col,
  clustering_method = "ward.D2",
  cluster_rows=FALSE,
  cluster_cols = TRUE,
  show_colnames=T,
  fontsize = 6.5
)
```




```
write.table(mat_hm, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/9_
    EGF_TC/2_Genelevel_analysis/RNA-Seq_egftc_heatmap_RIevents.txt", sep = "\t", row.names = TRUE
    , quote = F)
```

```
#Save the heatmap as pdf
pheatmap(mat_hm_scaled,
  main=NA,
  color = my.colors,
  breaks = my.breaks,
  annotation_col = annotation_col,
  clustering_method = "ward.D2",
  cluster_rows=FALSE,
  cluster_cols = TRUE,
  show_colnames=T,
  fontsize = 6.5,
  filename = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/siRNA_EGF_SG/2_EGF/9_EGF_TC/2_
    Genelevel_analysis/RNA-Seq_egftc_heatmap_RI.pdf",
  width = 3,
  height = 2.5
)
```

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      stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] lubridate_1.9.3             stringr_1.5.1
##  [3] dplyr_1.1.4                 purrr_1.0.2
##  [5] readr_2.1.5                 tidyr_1.3.1
##  [7] tibble_3.2.1                tidyverse_2.0.0
##  [9] SuperExactTest_1.1.0        eulerr_7.0.1
## [11] org.Hs.eg.db_3.15.0         viridis_0.6.5
## [13] viridisLite_0.4.2           gprofiler2_0.2.3
## [15] tximport_1.24.0             GenomicFeatures_1.48.4
## [17] AnnotationDbi_1.60.2        tximportData_1.24.0
## [19] forcats_1.0.0               ggExtra_0.10.1
## [21] ggpubr_0.6.0                MASS_7.3-60.0.1
## [23] ggplot2_3.5.0               pheatmap_1.0.12
## [25] DESeq2_1.38.3               SummarizedExperiment_1.28.0
## [27] Biobase_2.58.0              MatrixGenerics_1.10.0
## [29] matrixStats_1.2.0           GenomicRanges_1.50.2
## [31] GenomeInfoDb_1.34.9         IRanges_2.32.0
## [33] S4Vectors_0.36.2            BiocGenerics_0.44.0
##
## loaded via a namespace (and not attached):
##   [1] colorspace_2.1-0          ggsignif_0.6.4            rjson_0.2.21
##   [4] ellipsis_0.3.2            XVector_0.38.0            rstudioapi_0.15.0
##   [7] farver_2.1.1              ggrepel_0.9.5             bit64_4.0.5
##  [10] fansi_1.0.6               xml2_1.3.6                codetools_0.2-19
##  [13] cachem_1.0.8              geneplotter_1.76.0        knitr_1.45
##  [16] jsonlite_1.8.8            Rsamtools_2.14.0          broom_1.0.5
##  [19] annotate_1.76.0           dbplyr_2.4.0              png_0.1-8
##  [22] shiny_1.8.0               compiler_4.2.1            httr_1.4.7
##  [25] backports_1.4.1           lazyeval_0.2.2            Matrix_1.6-5
##  [28] fastmap_1.1.1             cli_3.6.2                 later_1.3.2
##  [31] htmltools_0.5.7           prettyunits_1.2.0         tools_4.2.1
##  [34] gtable_0.3.4              glue_1.7.0                GenomeInfoDbData_1.2.9
##  [37] rappdirs_0.3.3            Rcpp_1.0.12               carData_3.0-5
##  [40] vctrs_0.6.5               Biostrings_2.66.0         rtracklayer_1.56.1
##  [43] xfun_0.42                 timechange_0.3.0          mime_0.12
##  [46] miniUI_0.1.1.1            lifecycle_1.0.4           restfulr_0.0.15
##  [49] rstatix_0.7.2             XML_3.99-0.16.1           zlibbioc_1.44.0
##  [52] scales_1.3.0              vroom_1.6.5               hms_1.1.3
##  [55] promises_1.2.1            parallel_4.2.1            RColorBrewer_1.1-3
##  [58] yaml_2.3.8                curl_5.2.1                gridExtra_2.3
##  [61] memoise_2.0.1             biomaRt_2.52.0            stringi_1.8.3
##  [64] RSQLite_2.3.5             highr_0.10                BiocIO_1.6.0
##  [67] filelock_1.0.3            BiocParallel_1.32.6       rlang_1.1.3
##  [70] pkgconfig_2.0.3           bitops_1.0-7              evaluate_0.23
##  [73] lattice_0.22-5            labeling_0.4.3            htmlwidgets_1.6.4
##  [76] GenomicAlignments_1.34.1  bit_4.0.5                 tidyselect_1.2.1
##  [79] magrittr_2.0.3            R6_2.5.1                  generics_0.1.3
##  [82] DelayedArray_0.24.0       DBI_1.2.2                 pillar_1.9.0
```

```
##  [85] withr_3.0.0              KEGGREST_1.38.0          abind_1.4-5
##  [88] RCurl_1.98-1.14          crayon_1.5.2             car_3.1-2
##  [91] utf8_1.2.4               BiocFileCache_2.4.0      plotly_4.10.4
##  [94] tzdb_0.4.0               rmarkdown_2.26           progress_1.2.3
##  [97] locfit_1.5-9.9           data.table_1.15.2        blob_1.2.4
## [100] digest_0.6.35            xtable_1.8-4             httpuv_1.6.14
## [103] munsell_0.5.0
```