irCLIP-RNP dataset of species mixing experiment to determine contribution of mouse/human unique peptides

Luca Ducoli

28 June, 2024

This is the pipeline used to analyze the irCLIP-RNP datasets of species-mixing experiement. We have subjected to MS two gel sections ranging from 30 to 70kDa (named "free RNA ligation zone") and from 70 to 350KDa (named "whole RNP zone"). irCLIP-RNP for HNRNPC was performed after mixing UVC human cells (HEK293T) and noUV mouse cells (3T3s). irCLIP-RNP for HNRNPA2B1 was performed with noUV HEK293T and UVC 3T3 cells.

1. Prepare the dataset

```
#Load the libraries
library(trackViewer)
library(dplyr)
library(httr)
library(org.Hs.eg.db)
library(org.Mm.eg.db)
library(paletteer)
library(ggplot2)
library(data.table)
library(ggExtra)
library(DEP2)
library(DESeq2)
library(ggpubr)
```

In the first step, we prepared the dataset to create a SummarizedExperiment object starting from the peptide.txt output file from MaxQuant.

```
# Load the peptides
data <- read.csv("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/irCLIP-RNP_control/1_Species_
    mixing/0_Data/peptides.txt",
    sep = "\t")

# Remove RPL proteins
data <- data[-grep("RPL", data$Gene.names), ]

# Remove not detected peptides
data$rowsum <- rowSums(data[, grep("Intensity.", colnames(data),
    value = TRUE)])
data <- subset(data, rowsum != 0)
```

2. Create a SummarizedExperiment

We used the following design to create a SummarizedExperiment.

```
#Load design matrix
design <- read.delim("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/irCLIP-RNP_control/1_
    Species_mixing/0_Data/Design_matrix.txt")
design
```

```
##                 label      condition replicate
## 1 LFQ.intensity.BZ15A     HNRNPC_low         1
## 2 LFQ.intensity.BZ15B    HNRNPC_high         1
## 3 LFQ.intensity.BZ16A     HNRNPC_low         2
## 4 LFQ.intensity.BZ16B    HNRNPC_high         2
## 5 LFQ.intensity.BZ17A  HNRNPA2B1_low         1
## 6 LFQ.intensity.BZ17B HNRNPA2B1_high         1
## 7 LFQ.intensity.BZ18A  HNRNPA2B1_low         2
## 8 LFQ.intensity.BZ18B HNRNPA2B1_high         2
```

```
#Create a summarized experiment, filter, normalize, and impute
data$ID <- c(1:nrow(data))
data$name <- paste(data$Gene.names, data$ID, sep = "_")
ecols <- grep("Intensity.", colnames(data), value = TRUE)
pe <- make_pe(data, columns = ecols, expdesign = design)

#Filter, normalize, and impute
pe = filter_pe(pe, filter_formula = ~ Reverse != '+' & Potential.contaminant !="+" & Unique..
    Proteins. == "yes")
pe <- normalize_pe(pe,method = "vsn", i = "peptideRaw")
set.seed(13)
pe <- impute_pe(pe, fun = "GSimp", name = "peptideImp", i = "peptideNorm")
```

```
## Iteration 1 start...end!
## Iteration 2 start...end!
## Iteration 3 start...end!
## Iteration 4 start...end!
## Iteration 5 start...end!
## Iteration 6 start...end!
## Iteration 7 start...end!
## Iteration 8 start...end!
## Iteration 9 start...end!
## Iteration 10 start...end!
```

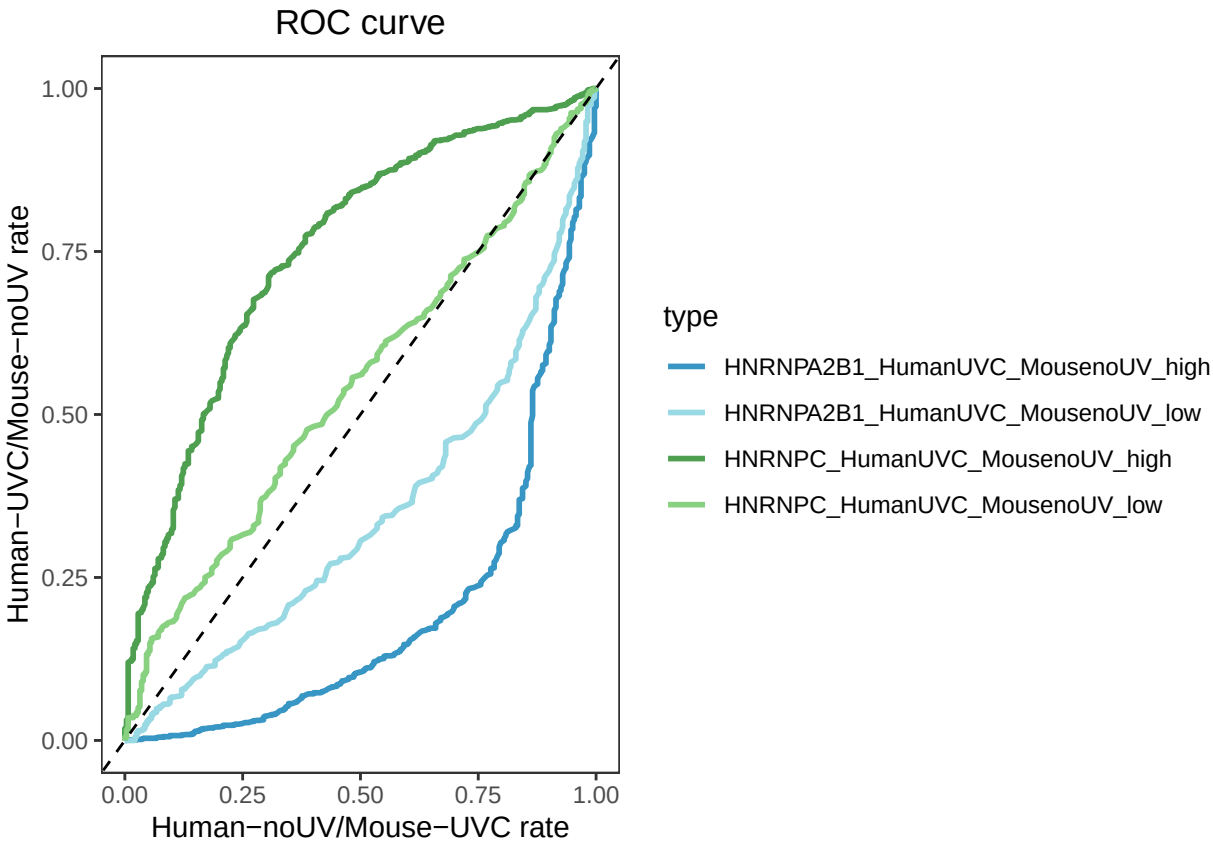
3. ROC analysis

We first performed a ROC analysis to compare UVC-human/noUV-mouse and noUV-human/UVC-mouse samples from the two gel sections.

```
#Prepare data.frame for ROC analysis
unique_pep <- assay(pe[[3]])
unique_pep <- as.data.frame(unique_pep)
rownames(unique_pep) <- pe[[3]]@elementMetadata$name

#Average between samples
```

```
unique_pep$HNRNPC_low <- rowMeans(unique_pep[c(1,3)])
unique_pep$HNRNPC_high <- rowMeans(unique_pep[c(2,4)])
unique_pep$HNRNPA2B1_low <- rowMeans(unique_pep[c(5,7)])
unique_pep$HNRNPA2B1_high <- rowMeans(unique_pep[c(6,8)])

#Subset mouse unique peptides
mouse_unique_pep <- unique_pep[grepl("[a-z]", rownames(unique_pep)), ]
mouse_unique_pep$Gene.names <- rownames(mouse_unique_pep)

mouse_HNRNPClow <- mouse_unique_pep[,c(13,9)]
colnames(mouse_HNRNPClow) <- c("Gene.names", "Int")

mouse_HNRNPChigh <- mouse_unique_pep[,c(13,10)]
colnames(mouse_HNRNPChigh) <- c("Gene.names", "Int")

mouse_HNRNPA2B1low <- mouse_unique_pep[,c(13,11)]
colnames(mouse_HNRNPA2B1low) <- c("Gene.names", "Int")

mouse_HNRNPA2B1high <- mouse_unique_pep[,c(13,12)]
colnames(mouse_HNRNPA2B1high) <- c("Gene.names", "Int")

#Subset human unique peptides
'%!in%' <- function(x,y)!('%in%'(x,y))
human_unique_pep <- subset(unique_pep, rownames(unique_pep) %!in% rownames(mouse_unique_pep))
human_unique_pep$Gene.names <- rownames(human_unique_pep)
human_HNRNPClow <- human_unique_pep[,c(13,9)]
colnames(human_HNRNPClow) <- c("Gene.names", "Int")
human_HNRNPChigh <- human_unique_pep[,c(13,10)]
colnames(human_HNRNPChigh) <- c("Gene.names", "Int")
human_HNRNPA2B1low <- human_unique_pep[,c(13,11)]
colnames(human_HNRNPA2B1low) <- c("Gene.names", "Int")
human_HNRNPA2B1high <- human_unique_pep[,c(13,12)]
colnames(human_HNRNPA2B1high) <- c("Gene.names", "Int")

#Perform cumulative fraction analysis
i <- seq(0, 30, by = 0.01)

mouse_ECDF_HNRNPChigh <- data.frame(seq = i, ecdf = 1-(ecdf(mouse_HNRNPChigh$Int)(i)))
colnames(mouse_ECDF_HNRNPChigh) <- c("Seq_m", "ROC_m")
human_ECDF_HNRNPChigh <- data.frame(seq = i, ecdf = 1-(ecdf(human_HNRNPChigh$Int)(i)))
colnames(human_ECDF_HNRNPChigh) <- c("Seq_h", "ROC_h")
HNRNPChigh <- cbind(human_ECDF_HNRNPChigh, mouse_ECDF_HNRNPChigh)
HNRNPChigh$type <- "HNRNPC_HumanUVC_MousenoUV_high"

mouse_ECDF_HNRNPClow <- data.frame(seq = i, ecdf = 1-(ecdf(mouse_HNRNPClow$Int)(i)))
colnames(mouse_ECDF_HNRNPClow) <- c("Seq_m", "ROC_m")
human_ECDF_HNRNPClow <- data.frame(seq = i, ecdf = 1-(ecdf(human_HNRNPClow$Int)(i)))
colnames(human_ECDF_HNRNPClow) <- c("Seq_h", "ROC_h")
HNRNPClow <- cbind(human_ECDF_HNRNPClow, mouse_ECDF_HNRNPClow)
HNRNPClow$type <- "HNRNPC_HumanUVC_MousenoUV_low"

mouse_ECDF_HNRNPA2B1high <- data.frame(seq = i, ecdf = 1-(ecdf(mouse_HNRNPA2B1high$Int)(i)))
colnames(mouse_ECDF_HNRNPA2B1high) <- c("Seq_m", "ROC_m")
human_ECDF_HNRNPA2B1high <- data.frame(seq = i, ecdf = 1-(ecdf(human_HNRNPA2B1high$Int)(i)))
colnames(human_ECDF_HNRNPA2B1high) <- c("Seq_h", "ROC_h")
HNRNPA2B1high <- cbind(human_ECDF_HNRNPA2B1high, mouse_ECDF_HNRNPA2B1high)
HNRNPA2B1high$type <- "HNRNPA2B1_HumanUVC_MousenoUV_high"

mouse_ECDF_HNRNPA2B1low <- data.frame(seq = i, ecdf = 1-(ecdf(mouse_HNRNPA2B1low$Int)(i)))
colnames(mouse_ECDF_HNRNPA2B1low) <- c("Seq_m", "ROC_m")
human_ECDF_HNRNPA2B1low <- data.frame(seq = i, ecdf = 1-(ecdf(human_HNRNPA2B1low$Int)(i)))
colnames(human_ECDF_HNRNPA2B1low) <- c("Seq_h", "ROC_h")
HNRNPA2B1low <- cbind(human_ECDF_HNRNPA2B1low, mouse_ECDF_HNRNPA2B1low)
HNRNPA2B1low$type <- "HNRNPA2B1_HumanUVC_MousenoUV_low"

#Create a dataframe for plotting
roc <- rbind(HNRNPChigh, HNRNPClow, HNRNPA2B1high, HNRNPA2B1low)
```

```
ggplot(roc %>% arrange(ROC_h, ROC_m), aes(x = ROC_m, y = ROC_h, color = type)) + geom_line(
    linewidth = 1) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed") +
  scale_color_manual(values=c("#3896C4", "#98D9E4","#4E9F50", "#87D180")) +
  theme_bw() +
  theme(panel.grid.major = element_blank(), #legend.position = "none",
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        axis.line = element_blank(),
        plot.title = element_text(hjust = 0.5)) +
  ggtitle("ROC curve") +
  ylab("Human-UVC/Mouse-noUV rate") +
  xlab("Human-noUV/Mouse-UVC rate")
```




```
#Save ROC curve as pdf
pdf("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/irCLIP-RNP_control/1_Species_mixing/2_ROC/
    ROC_MouseHuman.pdf", width = 5, height = 5)
ggplot(roc %>% arrange(ROC_h, ROC_m), aes(x = ROC_m, y = ROC_h, color = type)) + geom_line(
    linewidth = 1) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed") +
  scale_color_manual(values=c("#3896C4", "#98D9E4","#4E9F50", "#87D180")) +
  theme_bw() +
  theme(panel.grid.major = element_blank(), legend.position = "none",
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        axis.line = element_blank(),
        plot.title = element_text(hjust = 0.5)) +
  ggtitle("ROC curve") +
  ylab("Human-UVC/Mouse-noUV rate") +
  xlab("Human-noUV/Mouse-UVC rate")
dev.off()
```

4. Distribution of mouse/Human unique peptide accross RBPs

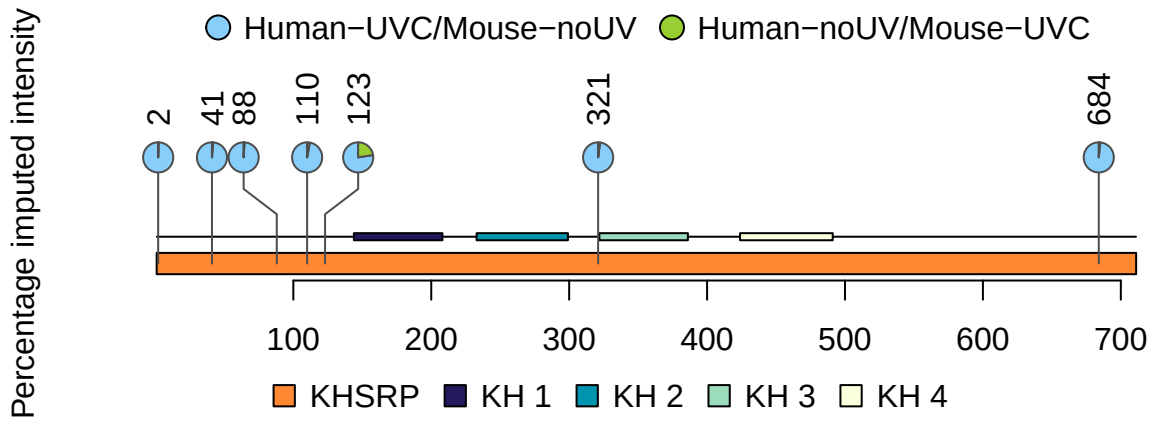
At this point, we displayed the ratio of imputed intensity of mouse/human unique peptides across some RBPs in the Human-UVC/Mouse-noUV and Human-noUV/Mouse-UVC samples. We used only the intensities coming from the gel section referred here as "high" (a.k.a. whole RNP zone) ranging from 70-350kDa.

```
#Function to draw the lolliplot
get_lolliplot <- function(gene, species, taxid, orgDB){
APIurl <- "https://www.ebi.ac.uk/proteins/api/"
eid <- mget(gene, get(sub(".db", "SYMBOL2EG", orgDB)))[[1]]
chr <- mget(eid, get(sub(".db", "CHR", orgDB)))[[1]]
accession <- unlist(lapply(eid, function(.ele){mget(.ele, get(sub(".db", "UNIPROT", orgDB)))}))
featureURL <- paste0(APIurl, "features?offset=0&size=-1&reviewed=true","&types=DNA_BIND%2CMOTIF%2
    CDOMAIN", "&taxid=", taxid,
                     "&accession=", paste(accession, collapse = "%2C"))
response <- GET(featureURL)
content <- httr::content(response)
content <- content[[1]]
acc <- content$accession
sequence <- content$sequence
gr <- GRanges(chr, IRanges(1, nchar(sequence)))
domains <- do.call(rbind, content$features)
domains <- GRanges(chr, IRanges(as.numeric(domains[, "begin"]),as.numeric(domains[, "end"]),names
     = domains[, "description"]))
domains$fill <- 1+seq_along(domains)
domains$height <- 0.04
fasta = readLines(paste("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/irCLIP-RNP_control/1_
    Species_mixing/3_Intensity_plot/", gene, "_", species, "_sequence.fasta", sep = ""))
ids = grepl(">", fasta)
rbp.f = data.frame(id = sub(">", "", fasta[ids]), s = tapply(fasta[!ids], cumsum(ids)[!ids],
    function(x) {paste(x, collapse = "")}))
rbp <- unique_pep[rownames(unique_pep) %like% gene, ]
rbp$name <- rownames(rbp)
rbp <- merge(rbp, data[c(37,84)], by = "name")
SNP <- sort(rbp$Start.position)

sample.gr <- GRanges(as.character(seqnames(domains)@values), IRanges(SNP, width=1, names=paste0(
    SNP)))
total_rbp_int <- (2^((rbp$HNRNPC_high_1+rbp$HNRNPC_high_2)/2)+2^((rbp$HNRNPA2B1_high_1+rbp$
    HNRNPA2B1_high_2)/2))

sample.gr$value1 <- 2^((rbp$HNRNPC_high_1+rbp$HNRNPC_high_2)/2)/total_rbp_int
sample.gr$value2 <- 2^((rbp$HNRNPA2B1_high_1+rbp$HNRNPA2B1_high_2)/2)/total_rbp_int
sample.gr$color <- rep(list(c("#87CEFA", '#98CE31')), length(SNP))
sample.gr$border <- "gray30"

features <- GRanges(as.character(seqnames(domains)@values), IRanges(c(1), width=nchar(rbp.f$s),
    names=paste0("block", 1)))
features$height <- c(0.03)

legend <- list(labels=c("Human-UVC/Mouse-noUV", "Human-noUV/Mouse-UVC"), fill=c("#87CEFA", '#98
    CE31'))

features.mul <- c(features, domains)
features.mul$height[c(2:(length(domains)+1))] <- c(rep(0.01, length(domains)))
features.mul$fill <- c("#FF8833", c(paletteer_c("grDevices::YlGnBu", (length(features.mul)-1))))
features.mul$featureLayerID <- c("tx_1", paste("tx", rep(2, each=length(domains)), sep="_"))
names(features.mul) <- c(gene, names(domains))
lolliplot <- lolliplot(sample.gr, features.mul, legend = legend, ylab="Percentage imputed
    intensity", type="pie")
```

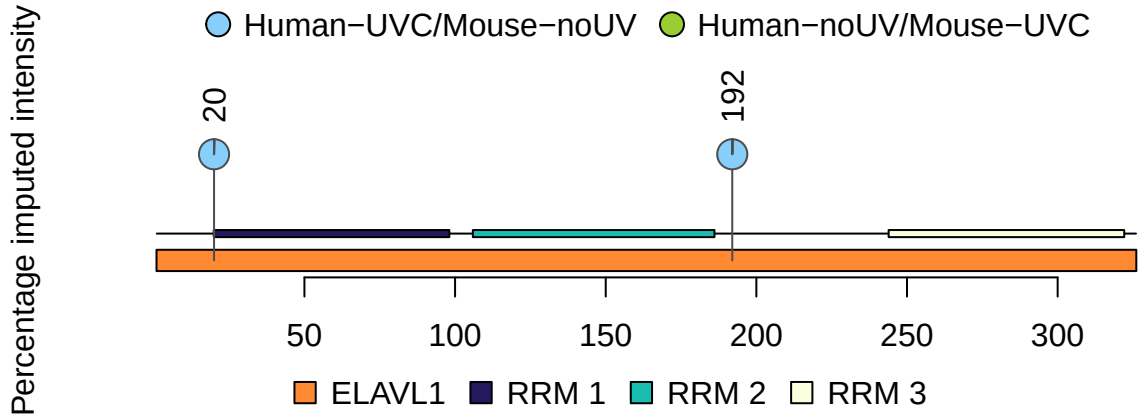
```
return(lolliplot)
}
```

Human RBPs

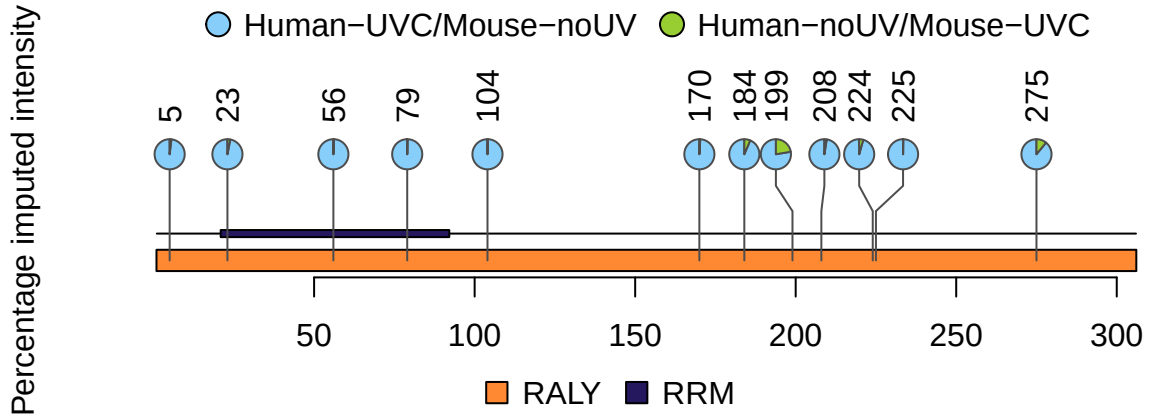
```
#Get lolliplot for human RBPs
KHSRP.h.lolliplot <- get_lolliplot("KHSRP", "Human", "9606","org.Hs.eg.db")
```



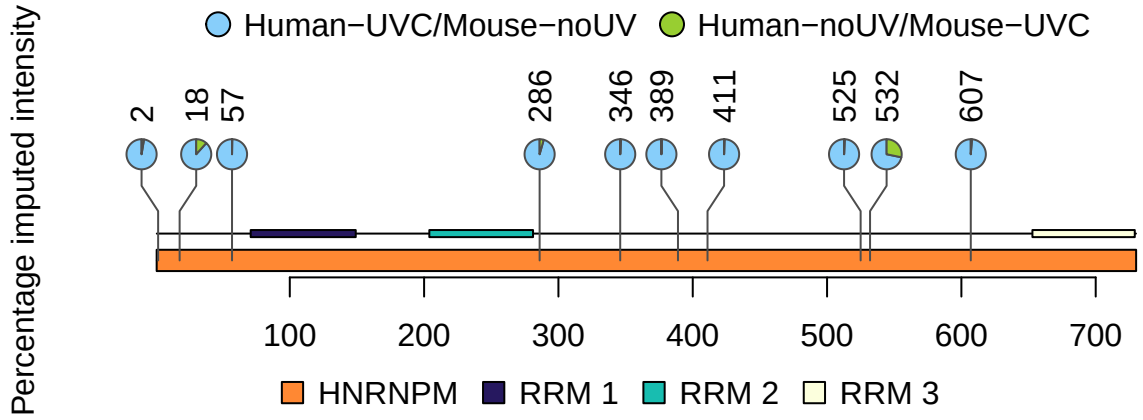

```
ELAVL1.h.lolliplot <- get_lolliplot("ELAVL1", "Human", "9606","org.Hs.eg.db")
```



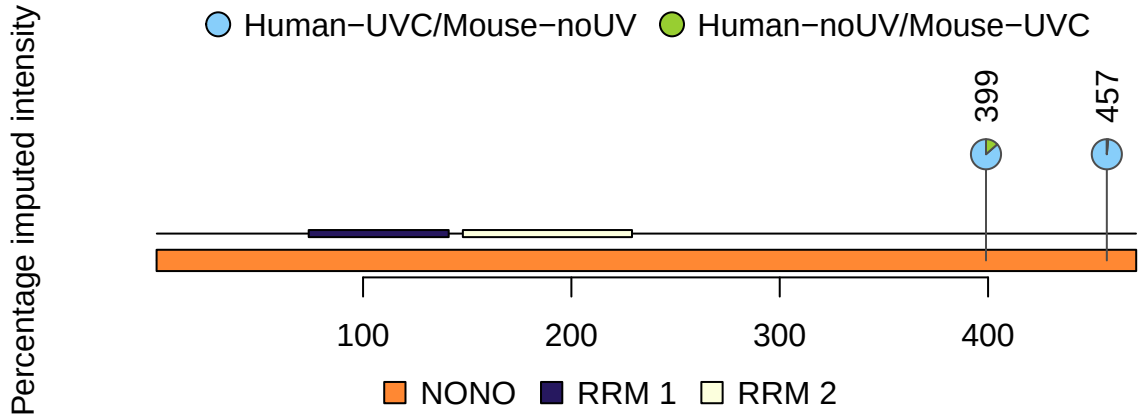

```
RALY.h.lolliplot <- get_lolliplot("RALY", "Human", "9606","org.Hs.eg.db")
```




```
HNRNPM.h.lolliplot <- get_lolliplot("HNRNPM", "Human", "9606","org.Hs.eg.db")
```



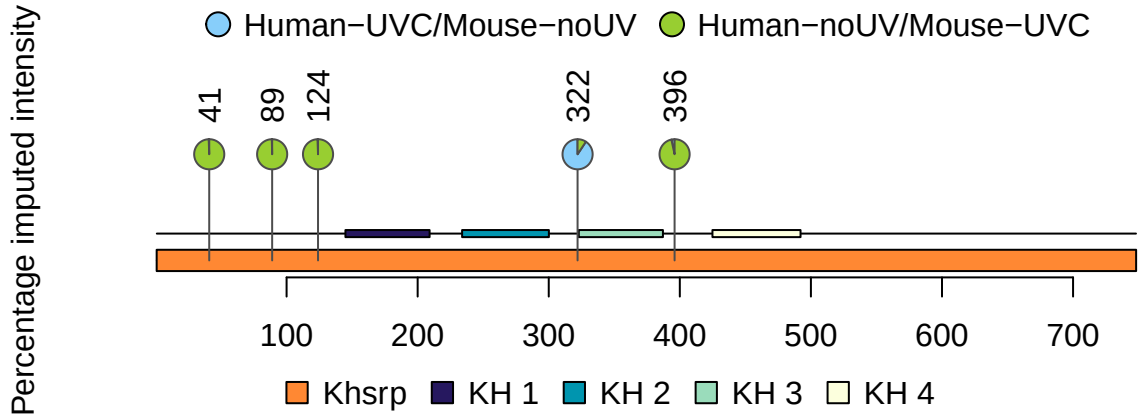

```
NONO.h.lolliplot <- get_lolliplot("NONO", "Human", "9606","org.Hs.eg.db")
```



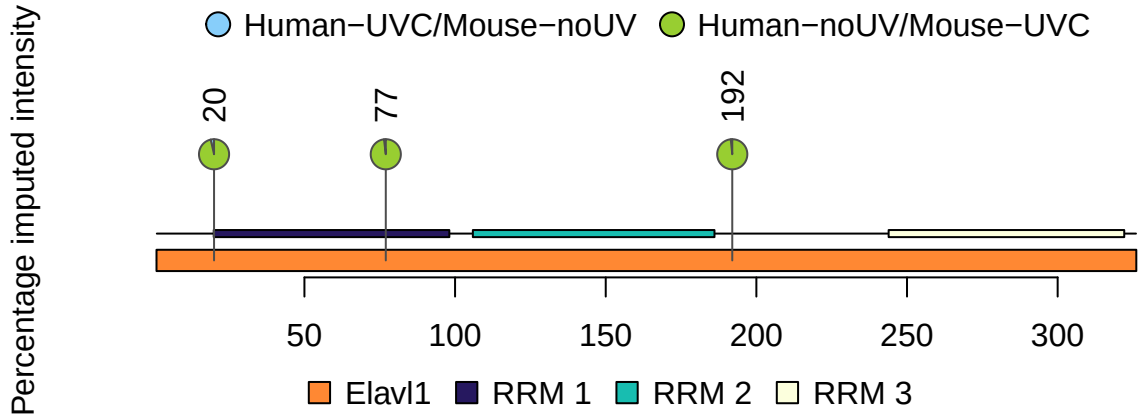

```
#Save human lolliplot as pdf
pdf("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/irCLIP-RNP_control/1_Species_mixing/3_
    Intensity_plot/Human_Lolliplot.pdf")
KHSRP.h.lolliplot <- get_lolliplot("KHSRP", "Human", "9606","org.Hs.eg.db")
ELAVL1.h.lolliplot <- get_lolliplot("ELAVL1", "Human", "9606","org.Hs.eg.db")
RALY.h.lolliplot <- get_lolliplot("RALY", "Human", "9606","org.Hs.eg.db")
HNRNPM.h.lolliplot <- get_lolliplot("HNRNPM", "Human", "9606","org.Hs.eg.db")
NONO.h.lolliplot <- get_lolliplot("NONO", "Human", "9606","org.Hs.eg.db")
dev.off()
```

Mouse RBPs

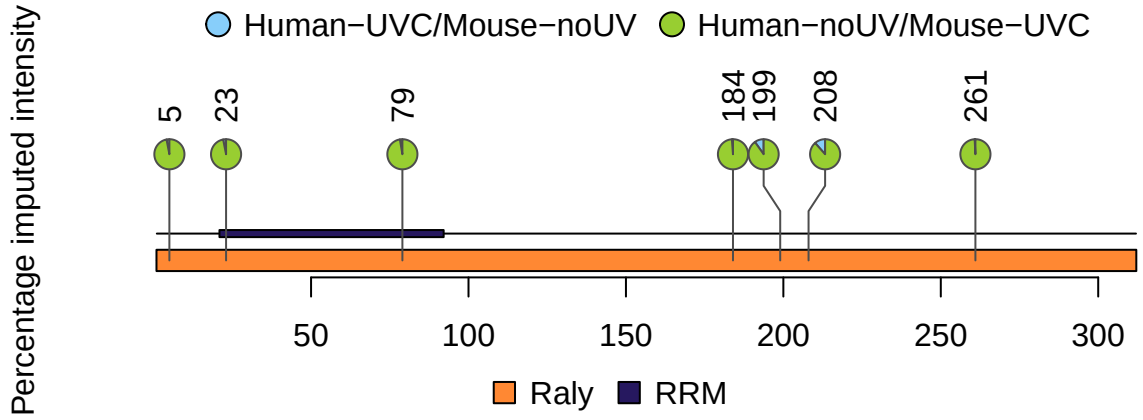
```
#Get lolliplot for mouse RBPs
Khsrp.m.lolliplot <- get_lolliplot("Khsrp", "Mouse", "10090", "org.Mm.eg.db")
```



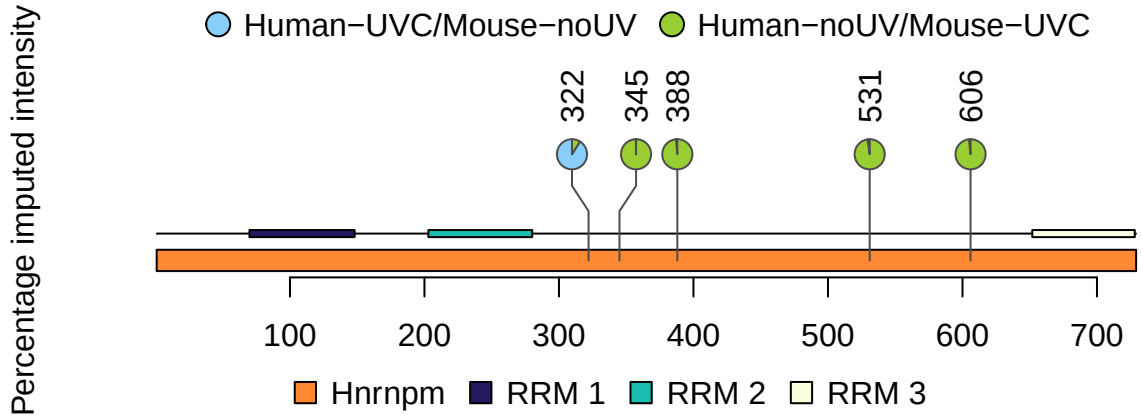

```
Elavl1.m.lolliplot <- get_lolliplot("Elavl1", "Mouse", "10090", "org.Mm.eg.db")
```



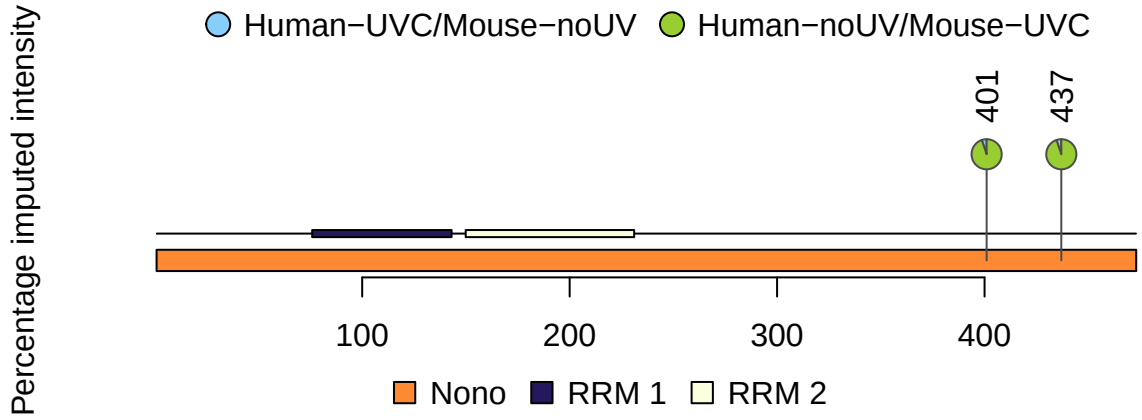

```
Raly.m.lolliplot <- get_lolliplot("Raly", "Mouse", "10090", "org.Mm.eg.db")
```




```
Hnrnpm.m.lolliplot <- get_lolliplot("Hnrnpm", "Mouse", "10090", "org.Mm.eg.db")
```




```
Nono.m.lolliplot <- get_lolliplot("Nono", "Mouse", "10090", "org.Mm.eg.db")
```




```
#Save mouse lolliplot as pdf
pdf("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/irCLIP-RNP_control/1_Species_mixing/3_
    Intensity_plot/Mouse_Lolliplot.pdf")
Khsrp.m.lolliplot <- get_lolliplot("Khsrp", "Mouse", "10090", "org.Mm.eg.db")
Elavl1.m.lolliplot <- get_lolliplot("Elavl1", "Mouse", "10090", "org.Mm.eg.db")
Raly.m.lolliplot <- get_lolliplot("Raly", "Mouse", "10090", "org.Mm.eg.db")
Hnrnpm.m.lolliplot <- get_lolliplot("Hnrnpm", "Mouse", "10090", "org.Mm.eg.db")
Nono.m.lolliplot <- get_lolliplot("Nono", "Mouse", "10090", "org.Mm.eg.db")
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      stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
```

```
##
## other attached packages:
##  [1] ggpubr_0.6.0                DESeq2_1.38.3
##  [3] DEP2_0.4.8.24               R6_2.5.1
##  [5] limma_3.54.2                MSnbase_2.24.2
##  [7] ProtGenerics_1.30.0         mzR_2.32.0
##  [9] MsCoreUtils_1.10.0          SummarizedExperiment_1.28.0
## [11] MatrixGenerics_1.10.0       matrixStats_1.2.0
## [13] ggExtra_0.10.1              data.table_1.15.2
## [15] ggplot2_3.5.0               paletteer_1.6.0
## [17] org.Mm.eg.db_3.15.0         org.Hs.eg.db_3.15.0
## [19] AnnotationDbi_1.60.2        Biobase_2.58.0
## [21] httr_1.4.7                  dplyr_1.1.4
## [23] trackViewer_1.32.1          Rcpp_1.0.12
## [25] GenomicRanges_1.50.2        GenomeInfoDb_1.34.9
## [27] IRanges_2.32.0              S4Vectors_0.36.2
## [29] BiocGenerics_0.44.0
##
## loaded via a namespace (and not attached):
##   [1] rappdirs_0.3.3               rtracklayer_1.56.1
##   [3] tidyr_1.3.1                  missForest_1.5
##   [5] bit64_4.0.5                  knitr_1.45
##   [7] DelayedArray_0.24.0          rpart_4.1.23
##   [9] KEGGREST_1.38.0              RCurl_1.98-1.14
##  [11] AnnotationFilter_1.22.0      doParallel_1.0.17
##  [13] generics_0.1.3               GenomicFeatures_1.48.4
##  [15] preprocessCore_1.60.2        RSQLite_2.3.5
##  [17] proxy_0.4-27                 bit_4.0.5
##  [19] xml2_1.3.6                   httpuv_1.6.14
##  [21] assertthat_0.2.1             TCseq_1.22.6
##  [23] xfun_0.42                    hms_1.1.3
##  [25] evaluate_0.23                promises_1.2.1
##  [27] fansi_1.0.6                  restfulr_0.0.15
##  [29] progress_1.2.3               dbplyr_2.4.0
##  [31] Rgraphviz_2.40.0             igraph_2.0.3
##  [33] DBI_1.2.2                    geneplotter_1.76.0
##  [35] htmlwidgets_1.6.4            purrr_1.0.2
##  [37] ellipsis_0.3.2               RSpectra_0.16-1
##  [39] QFeatures_1.8.0              backports_1.4.1
##  [41] prismatic_1.1.1              annotate_1.76.0
##  [43] biomaRt_2.52.0               deldir_2.0-4
##  [45] vctrs_0.6.5                  imputeLCMD_2.1
##  [47] ensembldb_2.20.2             abind_1.4-5
##  [49] cachem_1.0.8                 withr_3.0.0
##  [51] Gviz_1.40.1                  itertools_0.1-3
##  [53] BSgenome_1.64.0              checkmate_2.3.1
##  [55] GenomicAlignments_1.34.1     fdrtool_1.2.17
##  [57] prettyunits_1.2.0            MultiAssayExperiment_1.24.0
##  [59] cluster_2.1.6                BiocBaseUtils_1.0.0
##  [61] grImport_0.9-7               lazyeval_0.2.2
##  [63] crayon_1.5.2                 labeling_0.4.3
##  [65] glmnet_4.1-8                 edgeR_3.40.2
##  [67] pkgconfig_2.0.3              nnet_7.3-19
##  [69] rlang_1.1.3                  lifecycle_1.0.4
##  [71] miniUI_0.1.1.1               sandwich_3.1-0
##  [73] downloader_0.4               filelock_1.0.3
##  [75] affyio_1.68.0                BiocFileCache_2.4.0
##  [77] dichromat_2.0-0.1            randomForest_4.7-1.1
##  [79] graph_1.74.0                 rngtools_1.5.2
##  [81] Matrix_1.6-5                 carData_3.0-5
##  [83] zoo_1.8-12                   Rhdf5lib_1.20.0
##  [85] base64enc_0.1-3              GlobalOptions_0.1.2
##  [87] png_0.1-8                    rjson_0.2.21
##  [89] bitops_1.0-7                 rhdf5filters_1.10.1
##  [91] Biostrings_2.66.0            blob_1.2.4
##  [93] doRNG_1.8.6                  shape_1.4.6.1
##  [95] stringr_1.5.1                jpeg_0.1-10
##  [97] rstatix_0.7.2                tmvtnorm_1.6
```

```
##  [99] ggsignif_0.6.4                scales_1.3.0
## [101] memoise_2.0.1                 magrittr_2.0.3
## [103] plyr_1.8.9                    zlibbioc_1.44.0
## [105] compiler_4.2.1                BiocIO_1.6.0
## [107] RColorBrewer_1.1-3            plotrix_3.8-4
## [109] pcaMethods_1.90.0             clue_0.3-65
## [111] Rsamtools_2.14.0              cli_3.6.2
## [113] affy_1.76.0                   XVector_0.38.0
## [115] formatR_1.14                  htmlTable_2.4.2
## [117] Formula_1.2-5                 MASS_7.3-60.0.1
## [119] tidyselect_1.2.1              vsn_3.66.0
## [121] stringi_1.8.3                 highr_0.10
## [123] yaml_2.3.8                    norm_1.0-11.1
## [125] askpass_1.2.0                 locfit_1.5-9.9
## [127] latticeExtra_0.6-30           MALDIquant_1.22.2
## [129] VariantAnnotation_1.42.1      tools_4.2.1
## [131] parallel_4.2.1                circlize_0.4.16
## [133] rstudioapi_0.15.0             foreach_1.5.2
## [135] foreign_0.8-86                gridExtra_2.3
## [137] farver_2.1.1                  mzID_1.36.0
## [139] Rtsne_0.17                    digest_0.6.35
## [141] BiocManager_1.30.22           shiny_1.8.0
## [143] car_3.1-2                     broom_1.0.5
## [145] later_1.3.2                   ncdf4_1.22
## [147] biovizBase_1.44.0             ComplexHeatmap_2.14.0
## [149] colorspace_2.1-0              XML_3.99-0.16.1
## [151] reticulate_1.35.0             umap_0.2.10.0
## [153] splines_4.2.1                 rematch2_2.1.2
## [155] gmm_1.8                       xtable_1.8-4
## [157] jsonlite_1.8.8                Hmisc_5.1-2
## [159] pillar_1.9.0                  htmltools_0.5.7
## [161] mime_0.12                     glue_1.7.0
## [163] fastmap_1.1.1                 BiocParallel_1.32.6
## [165] class_7.3-22                  codetools_0.2-19
## [167] mvtnorm_1.2-4                 utf8_1.2.4
## [169] lattice_0.22-5                tibble_3.2.1
## [171] curl_5.2.1                    openssl_2.1.1
## [173] interp_1.1-6                  survival_3.5-8
## [175] rmarkdown_2.26                InteractionSet_1.24.0
## [177] munsell_0.5.0                 e1071_1.7-14
## [179] GetoptLong_1.0.5              rhdf5_2.42.1
## [181] GenomeInfoDbData_1.2.9        iterators_1.0.14
## [183] impute_1.72.3                 reshape2_1.4.4
## [185] gtable_0.3.4
```