irCLIP-RNP dataset from 13 RBPs in HEK293T and HepG2

Luca Ducoli

28 June, 2024

This is the pipeline used to analyze the label-free irCLIP-RNP datasets for 13 RBPs. The experiment consisted of 2 replicates UVC and 1 replicate no-UV samples derived from HEK293T and HepG2 cells.

1. Prepare the dataset

```
#Needed libraries
library(DEP2)
library(tidyverse)
library(ggplot2)
library(data.table)
library(pheatmap)
library(RColorBrewer)
library(gplots)
library(hrbrthemes)
library(pacman)
library(textshape)
library(ggExtra)
library(viridis)
library(purrr)
library(hexbin)
library(DESeq2)
library(ggpubr)
library(UpSetR)
library(dplyr)
library(Clipper)
library(factoextra)
library(paletteer)
library(corrplot)
library(psych)
library(ggpmisc)
library(gprofiler2)
library(viridis)
```

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

```
# Open proteinGroups.txt results from MaxQuant
data <- read.delim("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/Bulk_analysis/0_Data/
    proteinGroups.txt")

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

#Generate unique names and ids
unique_pg <- make_unique(data, name = "Gene.names", ids = "Protein.IDs")
unique_pg <- unique_pg %>% arrange(name)
```

```
#Get the columns
ecols <- grep("LFQ.intensity.", colnames(unique_pg))

#Keep isoform with higher LFQ intensity
iso <- grep("\\.\\d+$", unique_pg$name)
rbp <- gsub("\\.1", "", c(unique_pg$name[iso]))

#Find original row name of the isoform with higher intensity
find_max_value <- function(rbp) {
  filtered_df <- unique_pg[unique_pg$name %like% rbp, grep("LFQ.intensity.", colnames(unique_pg))
    ]
  filtered_df$rowSums <- rowSums(filtered_df[, grep("LFQ.intensity.", colnames(filtered_df))])
  max_value <- which.max(filtered_df$rowSums)
  rownames <- rownames(filtered_df)[-max_value]
  return(rownames)
}
max_iso <- c(unlist(lapply(rbp, find_max_value)))

#Remove low intensity isoforms
unique_pg <- unique_pg[!(rownames(unique_pg) %in% max_iso),]

# Remove all proteins detected in IgG
unique_pg <- subset(unique_pg, LFQ.intensity.BZ93 == 0 & LFQ.intensity.BZ94 == 0)

# Remove IgG column
unique_pg <- unique_pg[,-c(500,501)]
```

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/Bulk_analysis/0_Data/Design
    _matrix.txt")
design
```

```
##                label           condition cell_type replicate       rbp
## 1  LFQ.intensity.BZ10          ILF2_noUV      HepG2         2      ILF2
## 2  LFQ.intensity.BZ11     ILF2_UVC_HepG2      HepG2         1      ILF2
## 3  LFQ.intensity.BZ12     ILF2_UVC_HepG2      HepG2         2      ILF2
## 4  LFQ.intensity.BZ13          ILF3_noUV       293T         1      ILF3
## 5  LFQ.intensity.BZ14      ILF3_UVC_293T       293T         1      ILF3
## 6  LFQ.intensity.BZ15      ILF3_UVC_293T       293T         2      ILF3
## 7  LFQ.intensity.BZ16          ILF3_noUV      HepG2         2      ILF3
## 8  LFQ.intensity.BZ17     ILF3_UVC_HepG2      HepG2         1      ILF3
## 9  LFQ.intensity.BZ18     ILF3_UVC_HepG2      HepG2         2      ILF3
## 10 LFQ.intensity.BZ19           FUS_noUV       293T         1       FUS
## 11 LFQ.intensity.BZ20       FUS_UVC_293T       293T         1       FUS
## 12 LFQ.intensity.BZ21       FUS_UVC_293T       293T         2       FUS
## 13 LFQ.intensity.BZ22           FUS_noUV      HepG2         2       FUS
## 14 LFQ.intensity.BZ23      FUS_UVC_HepG2      HepG2         1       FUS
## 15 LFQ.intensity.BZ24      FUS_UVC_HepG2      HepG2         2       FUS
## 16 LFQ.intensity.BZ25          NONO_noUV       293T         1      NONO
## 17 LFQ.intensity.BZ26      NONO_UVC_293T       293T         1      NONO
## 18 LFQ.intensity.BZ27      NONO_UVC_293T       293T         2      NONO
## 19 LFQ.intensity.BZ28          NONO_noUV      HepG2         2      NONO
## 20 LFQ.intensity.BZ29     NONO_UVC_HepG2      HepG2         1      NONO
## 21 LFQ.intensity.BZ30     NONO_UVC_HepG2      HepG2         2      NONO
## 22 LFQ.intensity.BZ31          SFPQ_noUV       293T         1      SFPQ
## 23 LFQ.intensity.BZ32      SFPQ_UVC_293T       293T         1      SFPQ
```

## 24	LFQ.intensity.BZ33	SFPQ_UVC_293T	293T	2	SFPQ
## 25	LFQ.intensity.BZ34	SFPQ_noUV	HepG2	2	SFPQ
## 26	LFQ.intensity.BZ35	SFPQ_UVC_HepG2	HepG2	1	SFPQ
## 27	LFQ.intensity.BZ36	SFPQ_UVC_HepG2	HepG2	2	SFPQ
## 28	LFQ.intensity.BZ37	DDX5_noUV	293T	1	DDX5
## 29	LFQ.intensity.BZ38	DDX5_UVC_293T	293T	1	DDX5
## 30	LFQ.intensity.BZ39	DDX5_UVC_293T	293T	2	DDX5
## 31	LFQ.intensity.BZ40	DDX5_noUV	HepG2	2	DDX5
## 32	LFQ.intensity.BZ41	DDX5_UVC_HepG2	HepG2	1	DDX5
## 33	LFQ.intensity.BZ42	DDX5_UVC_HepG2	HepG2	2	DDX5
## 34	LFQ.intensity.BZ43	ABCF1_noUV	293T	1	ABCF1
## 35	LFQ.intensity.BZ44	ABCF1_UVC_293T	293T	1	ABCF1
## 36	LFQ.intensity.BZ45	ABCF1_UVC_293T	293T	2	ABCF1
## 37	LFQ.intensity.BZ46	ABCF1_noUV	HepG2	2	ABCF1
## 38	LFQ.intensity.BZ47	ABCF1_UVC_HepG2	HepG2	1	ABCF1
## 39	LFQ.intensity.BZ48	ABCF1_UVC_HepG2	HepG2	2	ABCF1
## 40	LFQ.intensity.BZ49	NAT10_noUV	293T	1	NAT10
## 41	LFQ.intensity.BZ50	NAT10_UVC_293T	293T	1	NAT10
## 42	LFQ.intensity.BZ51	NAT10_UVC_293T	293T	2	NAT10
## 43	LFQ.intensity.BZ52	NAT10_noUV	HepG2	2	NAT10
## 44	LFQ.intensity.BZ53	NAT10_UVC_HepG2	HepG2	1	NAT10
## 45	LFQ.intensity.BZ54	NAT10_UVC_HepG2	HepG2	2	NAT10
## 46	LFQ.intensity.BZ62	HNRNPA2B1_noUV	293T	1	HNRNPA2B1
## 47	LFQ.intensity.BZ63	HNRNPA2B1_UVC_293T	293T	1	HNRNPA2B1
## 48	LFQ.intensity.BZ64	HNRNPA2B1_UVC_293T	293T	2	HNRNPA2B1
## 49	LFQ.intensity.BZ65	HNRNPA2B1_noUV	HepG2	2	HNRNPA2B1
## 50	LFQ.intensity.BZ66	HNRNPA2B1_UVC_HepG2	HepG2	1	HNRNPA2B1
## 51	LFQ.intensity.BZ67	HNRNPA2B1_UVC_HepG2	HepG2	2	HNRNPA2B1
## 52	LFQ.intensity.BZ68	HNRNPC_noUV	293T	1	HNRNPC
## 53	LFQ.intensity.BZ69	HNRNPC_UVC_293T	293T	1	HNRNPC
## 54	LFQ.intensity.BZ7	ILF2_noUV	293T	1	ILF2
## 55	LFQ.intensity.BZ70	HNRNPC_UVC_293T	293T	2	HNRNPC
## 56	LFQ.intensity.BZ71	HNRNPC_noUV	HepG2	2	HNRNPC
## 57	LFQ.intensity.BZ72	HNRNPC_UVC_HepG2	HepG2	1	HNRNPC
## 58	LFQ.intensity.BZ73	HNRNPC_UVC_HepG2	HepG2	2	HNRNPC
## 59	LFQ.intensity.BZ74	HNRNPM_noUV	293T	1	HNRNPM
## 60	LFQ.intensity.BZ75	HNRNPM_UVC_293T	293T	1	HNRNPM
## 61	LFQ.intensity.BZ76	HNRNPM_UVC_293T	293T	2	HNRNPM
## 62	LFQ.intensity.BZ77	HNRNPM_noUV	HepG2	2	HNRNPM
## 63	LFQ.intensity.BZ78	HNRNPM_UVC_HepG2	HepG2	1	HNRNPM
## 64	LFQ.intensity.BZ79	HNRNPM_UVC_HepG2	HepG2	2	HNRNPM
## 65	LFQ.intensity.BZ8	ILF2_UVC_293T	293T	1	ILF2
## 66	LFQ.intensity.BZ80	HNRNPU_noUV	293T	1	HNRNPU
## 67	LFQ.intensity.BZ81	HNRNPU_UVC_293T	293T	1	HNRNPU
## 68	LFQ.intensity.BZ82	HNRNPU_UVC_293T	293T	2	HNRNPU
## 69	LFQ.intensity.BZ83	HNRNPU_noUV	HepG2	2	HNRNPU
## 70	LFQ.intensity.BZ84	HNRNPU_UVC_HepG2	HepG2	1	HNRNPU
## 71	LFQ.intensity.BZ85	HNRNPU_UVC_HepG2	HepG2	2	HNRNPU
## 72	LFQ.intensity.BZ87	RBFOX2_noUV	293T	1	RBFOX2
## 73	LFQ.intensity.BZ88	RBFOX2_UVC_293T	293T	1	RBFOX2
## 74	LFQ.intensity.BZ89	RBFOX2_UVC_293T	293T	2	RBFOX2
## 75	LFQ.intensity.BZ9	ILF2_UVC_293T	293T	2	ILF2
## 76	LFQ.intensity.BZ90	RBFOX2_noUV	HepG2	2	RBFOX2
## 77	LFQ.intensity.BZ91	RBFOX2_UVC_HepG2	HepG2	1	RBFOX2
## 78	LFQ.intensity.BZ92	RBFOX2_UVC_HepG2	HepG2	2	RBFOX2

	original_sample_name	crosslinking
## 1	ILF2.noUV.HepG2	noUV
## 2	ILF2.UVC.HepG2.R1	UVC
## 3	ILF2.UVC.HepG2.R2	UVC
## 4	ILF3.noUV.293T	noUV
## 5	ILF3.UVC.293T.R1	UVC
## 6	ILF3.UVC.293T.R2	UVC
## 7	ILF3.noUV.HepG2	noUV
## 8	ILF3.UVC.HepG2.R1	UVC
## 9	ILF3.UVC.HepG2.R2	UVC
## 10	FUS.noUV.293T	noUV
## 11	FUS.UVC.293T.R1	UVC
## 12	FUS.UVC.293T.R2	UVC

## 13	FUS.noUV.HepG2	noUV
## 14	FUS.UVC.HepG2.R1	UVC
## 15	FUS.UVC.HepG2.R2	UVC
## 16	NONO.noUV.293T	noUV
## 17	NONO.UVC.293T.R1	UVC
## 18	NONO.UVC.293T.R2	UVC
## 19	NONO.noUV.HepG2	noUV
## 20	NONO.UVC.HepG2.R1	UVC
## 21	NONO.UVC.HepG2.R2	UVC
## 22	SFPQ.noUV.293T	noUV
## 23	SFPQ.UVC.293T.R1	UVC
## 24	SFPQ.UVC.293T.R2	UVC
## 25	SFPQ.noUV.HepG2	noUV
## 26	SFPQ.UVC.HepG2.R1	UVC
## 27	SFPQ.UVC.HepG2.R2	UVC
## 28	DDX5.noUV.293T	noUV
## 29	DDX5.UVC.293T.R1	UVC
## 30	DDX5.UVC.293T.R2	UVC
## 31	DDX5.noUV.HepG2	noUV
## 32	DDX5.UVC.HepG2.R1	UVC
## 33	DDX5.UVC.HepG2.R2	UVC
## 34	ABCF1.noUV.293T	noUV
## 35	ABCF1.UVC.293T.R1	UVC
## 36	ABCF1.UVC.293T.R2	UVC
## 37	ABCF1.noUV.HepG2	noUV
## 38	ABCF1.UVC.HepG2.R1	UVC
## 39	ABCF1.UVC.HepG2.R2	UVC
## 40	NAT10.noUV.293T	noUV
## 41	NAT10.UVC.293T.R1	UVC
## 42	NAT10.UVC.293T.R2	UVC
## 43	NAT10.noUV.HepG2	noUV
## 44	NAT10.UVC.HepG2.R1	UVC
## 45	NAT10.UVC.HepG2.R2	UVC
## 46	hnA2B1.noUV.293T	noUV
## 47	hnA2B1.UVC.293T.R1	UVC
## 48	hnA2B1.UVC.293T.R2	UVC
## 49	hnA2B1.noUV.HepG2	noUV
## 50	hnA2B1.UVC.HepG2.R1	UVC
## 51	hnA2B1.UVC.HepG2.R2	UVC
## 52	hnC.noUV.293T	noUV
## 53	hnC.UVC.293T.R1	UVC
## 54	ILF2.noUV.293T	noUV
## 55	hnC.UVC.293T.R2	UVC
## 56	hnC.noUV.HepG2	noUV
## 57	hnC.UVC.HepG2.R1	UVC
## 58	hnC.UVC.HepG2.R2	UVC
## 59	hnM.noUV.293T	noUV
## 60	hnM.UVC.293T.R1	UVC
## 61	hnM.UVC.293T.R2	UVC
## 62	hnM.noUV.HepG2	noUV
## 63	hnM.UVC.HepG2.R1	UVC
## 64	hnM.UVC.HepG2.R2	UVC
## 65	ILF2.UVC.293T.R1	UVC
## 66	hnU.noUV.293T	noUV
## 67	hnU.UVC.293T.R1	UVC
## 68	hnU.UVC.293T.R2	UVC
## 69	hnU.noUV.HepG2	noUV
## 70	hnU.UVC.HepG2.R1	UVC
## 71	hnU.UVC.HepG2.R2	UVC
## 72	RBFOX2.noUV.293T	noUV
## 73	RBFOX2.UVC.293T.R1	UVC
## 74	RBFOX2.UVC.293T.R2	UVC
## 75	ILF2.UVC.293T.R2	UVC
## 76	RBFOX2.noUV.HepG2	noUV
## 77	RBFOX2.UVC.HepG2.R1	UVC
## 78	RBFOX2.UVC.HepG2.R2	UVC

```
# Create a SummarizedExperiment
se <- make_se(unique_pg, columns = ecols, expdesign = design)
se_UVC <- se[,se$crosslinking == "UVC"]
se_UVC <- filter_se(se_UVC, thr = 0, filter_formula = ~ Reverse != '+' & Potential.contaminant !=
    "+" & Peptides > 1 & Unique.peptides > 0)
se <- se[rownames(se_UVC),]
write.table(as.data.frame(se@assays@data@listData), file = "~/Documents/Postdoc/PD_Projects/3_
    irCLIP-RNP/MS/Bulk_analysis/2_UVC_enriched/All_se_LFQ_intensity_raw.txt", row.names = TRUE,
    sep = "\t", quote = F)

#Subset se by cell line
HEK293T_se <- se[,se$cell_type == "293T"]
HepG2_se <- se[,se$cell_type == "HepG2"]

# Subset the se according to the proteins
ABCF1_HEK293T_se <- HEK293T_se[,HEK293T_se$rbp == "ABCF1"]
DDX5_HEK293T_se <- HEK293T_se[,HEK293T_se$rbp == "DDX5" ]
FUS_HEK293T_se <- HEK293T_se[,HEK293T_se$rbp == "FUS" ]
HNRNPA2B1_HEK293T_se <- HEK293T_se[,HEK293T_se$rbp == "HNRNPA2B1" ]
HNRNPC_HEK293T_se <- HEK293T_se[,HEK293T_se$rbp == "HNRNPC" ]
HNRNPM_HEK293T_se <- HEK293T_se[,HEK293T_se$rbp == "HNRNPM" ]
HNRNPU_HEK293T_se <- HEK293T_se[,HEK293T_se$rbp == "HNRNPU" ]
ILF2_HEK293T_se <- HEK293T_se[,HEK293T_se$rbp == "ILF2" ]
ILF3_HEK293T_se <- HEK293T_se[,HEK293T_se$rbp == "ILF3" ]
NAT10_HEK293T_se <- HEK293T_se[,HEK293T_se$rbp == "NAT10" ]
NONO_HEK293T_se <- HEK293T_se[,HEK293T_se$rbp == "NONO" ]
RBFOX2_HEK293T_se <- HEK293T_se[,HEK293T_se$rbp == "RBFOX2" ]
SFPQ_HEK293T_se <- HEK293T_se[,HEK293T_se$rbp == "SFPQ" ]

ABCF1_HepG2_se <- HepG2_se[,HepG2_se$rbp == "ABCF1"]
DDX5_HepG2_se <- HepG2_se[,HepG2_se$rbp == "DDX5" ]
FUS_HepG2_se <- HepG2_se[,HepG2_se$rbp == "FUS" ]
HNRNPA2B1_HepG2_se <- HepG2_se[,HepG2_se$rbp == "HNRNPA2B1" ]
HNRNPC_HepG2_se <- HepG2_se[,HepG2_se$rbp == "HNRNPC" ]
HNRNPM_HepG2_se <- HepG2_se[,HepG2_se$rbp == "HNRNPM" ]
HNRNPU_HepG2_se <- HepG2_se[,HepG2_se$rbp == "HNRNPU" ]
ILF2_HepG2_se <- HepG2_se[,HepG2_se$rbp == "ILF2" ]
ILF3_HepG2_se <- HepG2_se[,HepG2_se$rbp == "ILF3" ]
NAT10_HepG2_se <- HepG2_se[,HepG2_se$rbp == "NAT10" ]
NONO_HepG2_se <- HepG2_se[,HepG2_se$rbp == "NONO" ]
RBFOX2_HepG2_se <- HepG2_se[,HepG2_se$rbp == "RBFOX2" ]
SFPQ_HepG2_se <- HepG2_se[,HepG2_se$rbp == "SFPQ" ]
```

3. Perform FDR analysis using ClipperR

We next prepared the data for ClipperR analysis by comparing the noUV samples vs UVC for each cell line. We compared noUV (1 replicate) and UVC samples (2 replicates for each RBP in each cell line).

```
set.seed(3)
# Clipper function
clipper_f <- function (flt) {
  flt_temp <- flt[,flt$crosslinking == "UVC"]
  flt_temp <- filter_se(flt_temp, thr = 0)
  flt <- flt[rownames(flt_temp), ]
  imputed <- DEP2::impute(flt , fun = "QRILC")
  data <- as.data.frame(assay(imputed))
  clipper = Clipper(score.exp = as.matrix(data[,c(2,3)]), score.back = as.matrix(data[,-c(2,3)]),
     FDR = 0.05, analysis = "e")
  data$FDR <- clipper$q
  data <- cbind(data, rowMeans(data[,c(2,3)])-data[1])
  colnames(data)[5] <- c("logFC")
  deg <- subset(data, FDR < 0.1 & logFC > log2(3))
```

```
  return(list(imp = imputed, all = data, deg = deg))
}

# Run Clipper
ABCF1_HEK293T_clipper <- clipper_f(ABCF1_HEK293T_se)
DDX5_HEK293T_clipper <- clipper_f(DDX5_HEK293T_se)
FUS_HEK293T_clipper <- clipper_f(FUS_HEK293T_se)
HNRNPA2B1_HEK293T_clipper <- clipper_f(HNRNPA2B1_HEK293T_se)
HNRNPC_HEK293T_clipper <- clipper_f(HNRNPC_HEK293T_se)
HNRNPM_HEK293T_clipper <- clipper_f(HNRNPM_HEK293T_se)
HNRNPU_HEK293T_clipper <- clipper_f(HNRNPU_HEK293T_se)
ILF2_HEK293T_clipper <- clipper_f(ILF2_HEK293T_se)
ILF3_HEK293T_clipper <- clipper_f(ILF3_HEK293T_se)
NAT10_HEK293T_clipper <- clipper_f(NAT10_HEK293T_se)
NONO_HEK293T_clipper <- clipper_f(NONO_HEK293T_se)
RBFOX2_HEK293T_clipper <- clipper_f(RBFOX2_HEK293T_se)
SFPQ_HEK293T_clipper <- clipper_f(SFPQ_HEK293T_se)

ABCF1_HepG2_clipper <- clipper_f(ABCF1_HepG2_se)
DDX5_HepG2_clipper <- clipper_f(DDX5_HepG2_se)
FUS_HepG2_clipper <- clipper_f(FUS_HepG2_se)
HNRNPA2B1_HepG2_clipper <- clipper_f(HNRNPA2B1_HepG2_se)
HNRNPC_HepG2_clipper <- clipper_f(HNRNPC_HepG2_se)
HNRNPM_HepG2_clipper <- clipper_f(HNRNPM_HepG2_se)
HNRNPU_HepG2_clipper <- clipper_f(HNRNPU_HepG2_se)
ILF2_HepG2_clipper <- clipper_f(ILF2_HepG2_se)
ILF3_HepG2_clipper <- clipper_f(ILF3_HepG2_se)
NAT10_HepG2_clipper <- clipper_f(NAT10_HepG2_se)
NONO_HepG2_clipper <- clipper_f(NONO_HepG2_se)
RBFOX2_HepG2_clipper <- clipper_f(RBFOX2_HepG2_se)
SFPQ_HepG2_clipper <- clipper_f(SFPQ_HepG2_se)

head(HNRNPC_HEK293T_clipper$deg)
```

```
##          HNRNPC_noUV_1 HNRNPC_UVC_293T_1 HNRNPC_UVC_293T_2       FDR      logFC
## DHX15         17.833891           19.50731           20.17658 0.0212766  2.008051
## DHX9          20.888294           23.71555           23.90150 0.0212766  2.920231
## ELAVL1        11.584692           25.98028           26.02673 0.0212766 14.418817
## FAM120A        7.816042           20.72908           21.00251 0.0212766 13.049752
## FUBP1         15.913095           20.76207           21.07411 0.0212766  5.004995
## FUBP3         16.360316           22.36642           22.67696 0.0212766  6.161374
```

```
# Save the clipper results
write.table(ABCF1_HEK293T_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/
    Bulk_analysis/2_UVC_enriched/ABCF1_HEK293T_clipper_results.txt", row.names = TRUE, sep = "\t"
    )
write.table(DDX5_HEK293T_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/
    Bulk_analysis/2_UVC_enriched/DDX5_HEK293T_clipper_results.txt", row.names = TRUE, sep = "\t")
write.table(FUS_HEK293T_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/Bulk
    _analysis/2_UVC_enriched/FUS_HEK293T_clipper_results.txt", row.names = TRUE, sep = "\t")
write.table(HNRNPA2B1_HEK293T_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/
    MS/Bulk_analysis/2_UVC_enriched/HNRNPA2B1_HEK293T_clipper_results.txt", row.names = TRUE, sep
     = "\t")
write.table(HNRNPC_HEK293T_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/
    Bulk_analysis/2_UVC_enriched/HNRNPC_HEK293T_clipper_results.txt", row.names = TRUE, sep = "\t
    ")
write.table(HNRNPM_HEK293T_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/
    Bulk_analysis/2_UVC_enriched/HNRNPM_HEK293T_clipper_results.txt", row.names = TRUE, sep = "\t
    ")
write.table(HNRNPU_HEK293T_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/
    Bulk_analysis/2_UVC_enriched/HNRNPU_HEK293T_clipper_results.txt", row.names = TRUE, sep = "\t
    ")
write.table(ILF2_HEK293T_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/
    Bulk_analysis/2_UVC_enriched/ILF2_HEK293T_clipper_results.txt", row.names = TRUE, sep = "\t")
```

```
write.table(ILF3_HEK293T_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/
    Bulk_analysis/2_UVC_enriched/ILF3_HEK293T_clipper_results.txt", row.names = TRUE, sep = "\t")
write.table(NAT10_HEK293T_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/
    Bulk_analysis/2_UVC_enriched/NAT10_HEK293T_clipper_results.txt", row.names = TRUE, sep = "\t"
    )
write.table(NONO_HEK293T_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/
    Bulk_analysis/2_UVC_enriched/NONO_HEK293T_clipper_results.txt", row.names = TRUE, sep = "\t")
write.table(RBFOX2_HEK293T_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/
    Bulk_analysis/2_UVC_enriched/RBFOX2_HEK293T_clipper_results.txt", row.names = TRUE, sep = "\t
    ")
write.table(SFPQ_HEK293T_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/
    Bulk_analysis/2_UVC_enriched/SFPQ_HEK293T_clipper_results.txt", row.names = TRUE, sep = "\t")

write.table(ABCF1_HepG2_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/Bulk
    _analysis/2_UVC_enriched/ABCF1_HepG2_clipper_results.txt", row.names = TRUE, sep = "\t")
write.table(DDX5_HepG2_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/Bulk_
    analysis/2_UVC_enriched/DDX5_HepG2_clipper_results.txt", row.names = TRUE, sep = "\t")
write.table(FUS_HepG2_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/Bulk_
    analysis/2_UVC_enriched/FUS_HepG2_clipper_results.txt", row.names = TRUE, sep = "\t")
write.table(HNRNPA2B1_HepG2_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/
    Bulk_analysis/2_UVC_enriched/HNRNPA2B1_HepG2_clipper_results.txt", row.names = TRUE, sep = "\
    t")
write.table(HNRNPC_HepG2_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/
    Bulk_analysis/2_UVC_enriched/HNRNPC_HepG2_clipper_results.txt", row.names = TRUE, sep = "\t")
write.table(HNRNPM_HepG2_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/
    Bulk_analysis/2_UVC_enriched/HNRNPM_HepG2_clipper_results.txt", row.names = TRUE, sep = "\t")
write.table(HNRNPU_HepG2_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/
    Bulk_analysis/2_UVC_enriched/HNRNPU_HepG2_clipper_results.txt", row.names = TRUE, sep = "\t")
write.table(ILF2_HepG2_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/Bulk_
    analysis/2_UVC_enriched/ILF2_HepG2_clipper_results.txt", row.names = TRUE, sep = "\t")
write.table(ILF3_HepG2_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/Bulk_
    analysis/2_UVC_enriched/ILF3_HepG2_clipper_results.txt", row.names = TRUE, sep = "\t")
write.table(NAT10_HepG2_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/Bulk
    _analysis/2_UVC_enriched/NAT10_HepG2_clipper_results.txt", row.names = TRUE, sep = "\t")
write.table(NONO_HepG2_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/Bulk_
    analysis/2_UVC_enriched/NONO_HepG2_clipper_results.txt", row.names = TRUE, sep = "\t")
write.table(RBFOX2_HepG2_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/
    Bulk_analysis/2_UVC_enriched/RBFOX2_HepG2_clipper_results.txt", row.names = TRUE, sep = "\t")
write.table(SFPQ_HepG2_clipper$all, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/Bulk_
    analysis/2_UVC_enriched/SFPQ_HepG2_clipper_results.txt", row.names = TRUE, sep = "\t")
```

4. Visualization of the results

For the visualization, we first generated a list of all UVC-enriched proteins. Only proteins with an FDR < 0.1 and a FC vs noUV > 3 in at least one condition were categorized as UVC-enriched.

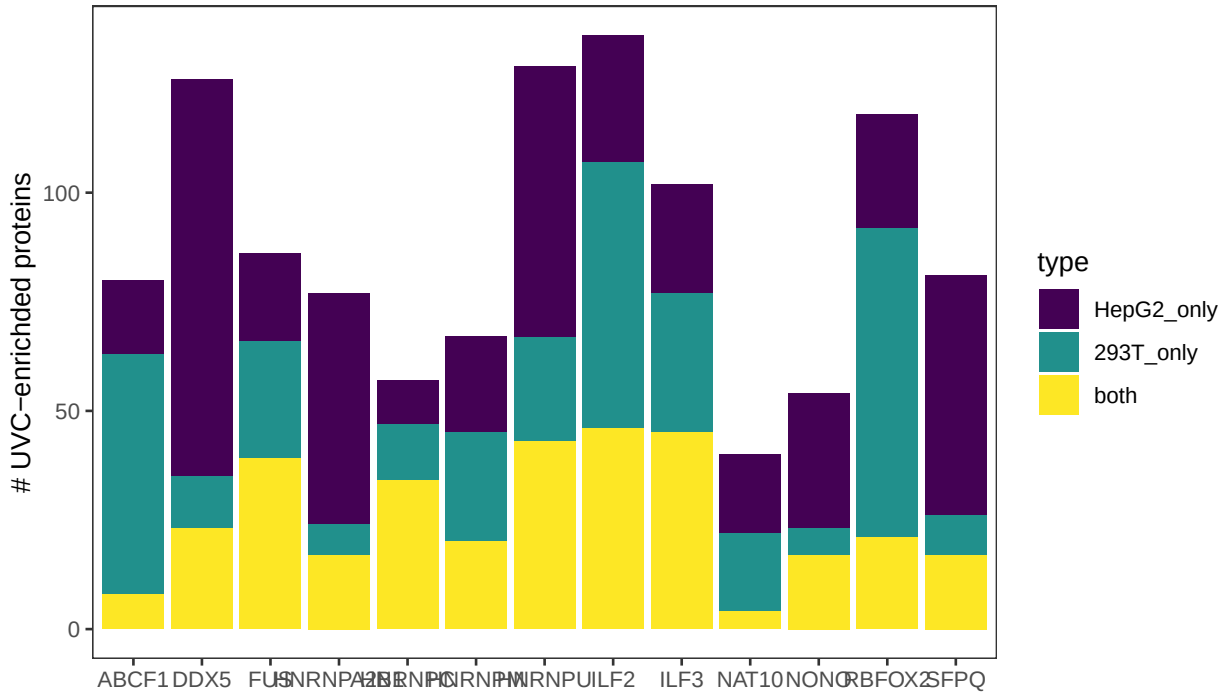
```
#Get total unique UVC proteins
UVC_all_HEK293T <- unique(c(rownames(ABCF1_HEK293T_clipper$deg), rownames(DDX5_HEK293T_clipper$
    deg), rownames(FUS_HEK293T_clipper$deg),
                            rownames(HNRNPA2B1_HEK293T_clipper$deg),rownames(HNRNPC_HEK293T_
    clipper$deg),rownames(HNRNPM_HEK293T_clipper$deg),
                            rownames(HNRNPU_HEK293T_clipper$deg), rownames(ILF2_HEK293T_clipper$
    deg), rownames(ILF3_HEK293T_clipper$deg),
                            rownames(NAT10_HEK293T_clipper$deg), rownames(NONO_HEK293T_clipper$
    deg), rownames(RBFOX2_HEK293T_clipper$deg),
                            rownames(SFPQ_HEK293T_clipper$deg)))

UVC_all_HepG2 <- unique(c(rownames(ABCF1_HepG2_clipper$deg), rownames(DDX5_HepG2_clipper$deg),
    rownames(FUS_HepG2_clipper$deg),
                          rownames(HNRNPA2B1_HepG2_clipper$deg),rownames(HNRNPC_HepG2_clipper$deg
    ), rownames(HNRNPM_HepG2_clipper$deg),
                          rownames(HNRNPU_HepG2_clipper$deg), rownames(ILF2_HepG2_clipper$deg),
    rownames(ILF3_HepG2_clipper$deg),
```

```
                                rownames(NAT10_HepG2_clipper$deg), rownames(NONO_HepG2_clipper$deg),
    rownames(RBFOX2_HepG2_clipper$deg),
                                rownames(SFPQ_HepG2_clipper$deg)))

UVC_all <- unique(c(UVC_all_HEK293T, UVC_all_HepG2))
rowData <- as.data.frame(rowData(se))
UVC_rowData <- rowData[UVC_all,]

write.table(UVC_rowData, file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/Bulk_analysis/2_
    UVC_enriched/UVC_proteins_All.txt", row.names = FALSE, sep = "\t", quote = FALSE)
```

Barplot of the UVC-enriched proteins between cell type

```
# Function to create the binary table
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)
}

# Generate a table how many of the UVC proteins are in common between cell types
celltype <- function (input, input2) {
list <- list(H293T = rownames(input),HepG2 =  rownames(input2))
list <- as.data.frame(fromList(list))
list$type <- factor(paste(list$H293T, list$HepG2, sep = "_"))
HEK293T <- rownames(subset(list, type %in% c("1_0", "1_1")))
HepG2 <- rownames(subset(list, type %in% c("0_1", "1_1")))
list <- list %>% dplyr::count(type)
return(list("celltype" = list, "HEK293T" = HEK293T, "HepG2" = HepG2))
}

ABCF1_list <- celltype(ABCF1_HEK293T_clipper$deg, ABCF1_HepG2_clipper$deg)
ABCF1_list$celltype$name <- "ABCF1"
DDX5_list <- celltype(DDX5_HEK293T_clipper$deg, DDX5_HepG2_clipper$deg)
DDX5_list$celltype$name <- "DDX5"
FUS_list <- celltype(FUS_HEK293T_clipper$deg, FUS_HepG2_clipper$deg)
FUS_list$celltype$name <- "FUS"
HNRNPA2B1_list <- celltype(HNRNPA2B1_HEK293T_clipper$deg, HNRNPA2B1_HepG2_clipper$deg)
HNRNPA2B1_list$celltype$name <- "HNRNPA2B1"
HNRNPC_list <- celltype(HNRNPC_HEK293T_clipper$deg, HNRNPC_HepG2_clipper$deg)
HNRNPC_list$celltype$name <- "HNRNPC"
HNRNPM_list <- celltype(HNRNPM_HEK293T_clipper$deg, HNRNPM_HepG2_clipper$deg)
HNRNPM_list$celltype$name <- "HNRNPM"
HNRNPU_list <- celltype(HNRNPU_HEK293T_clipper$deg, HNRNPU_HepG2_clipper$deg)
HNRNPU_list$celltype$name <- "HNRNPU"
ILF2_list <- celltype(ILF2_HEK293T_clipper$deg, ILF2_HepG2_clipper$deg)
ILF2_list$celltype$name <- "ILF2"
ILF3_list <- celltype(ILF3_HEK293T_clipper$deg, ILF3_HepG2_clipper$deg)
ILF3_list$celltype$name <- "ILF3"
NAT10_list <- celltype(NAT10_HEK293T_clipper$deg, NAT10_HepG2_clipper$deg)
NAT10_list$celltype$name <- "NAT10"
NONO_list <- celltype(NONO_HEK293T_clipper$deg, NONO_HepG2_clipper$deg)
NONO_list$celltype$name <- "NONO"
RBFOX2_list <- celltype(RBFOX2_HEK293T_clipper$deg, RBFOX2_HepG2_clipper$deg)
RBFOX2_list$celltype$name <- "RBFOX2"
```

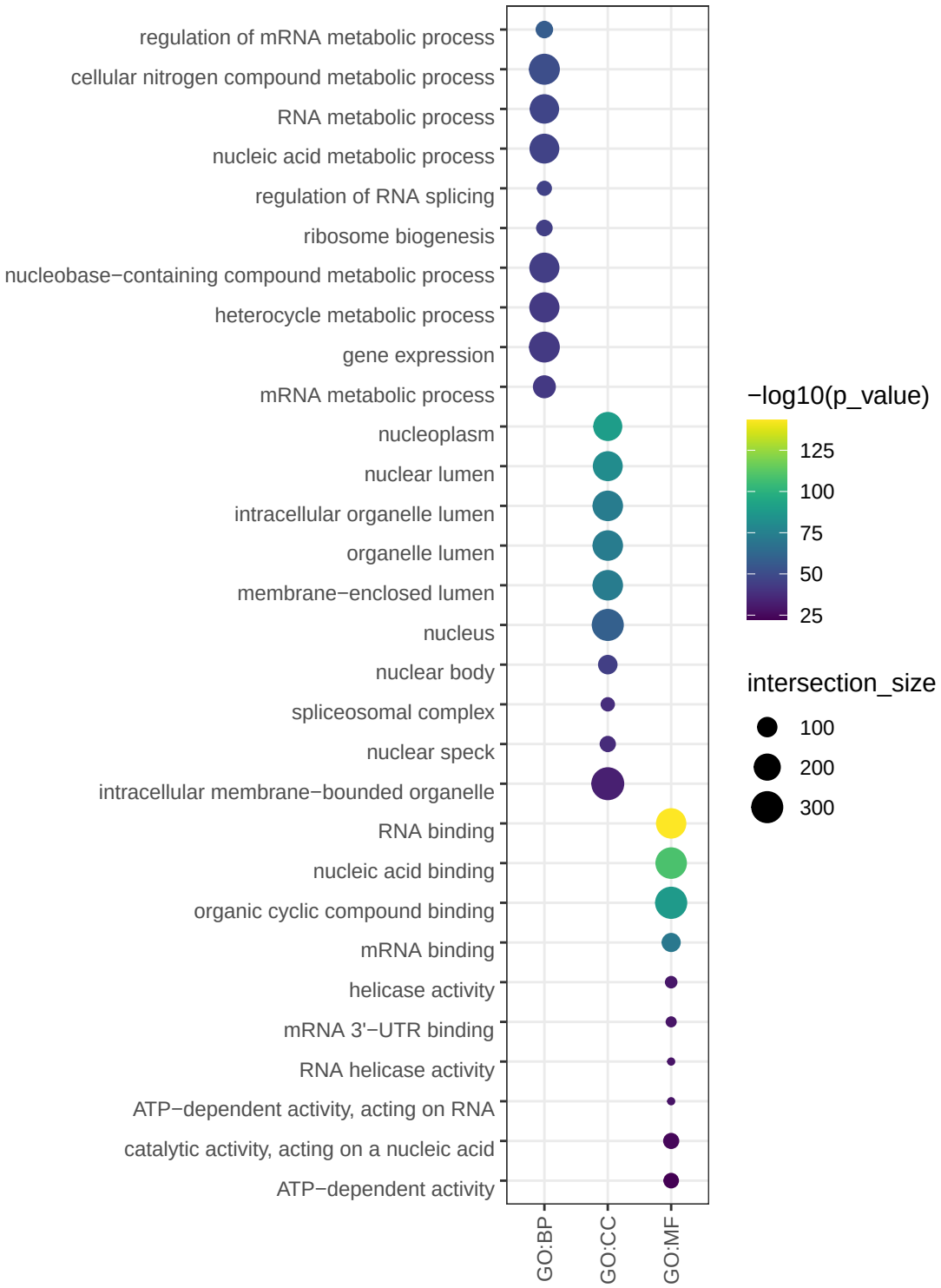
```
SFPQ_list <- celltype(SFPQ_HEK293T_clipper$deg, SFPQ_HepG2_clipper$deg)
SFPQ_list$celltype$name <- "SFPQ"

# Combine all results for each RBP to be used in ggplot
all <- rbind(ABCF1_list$celltype, DDX5_list$celltype, FUS_list$celltype, HNRNPA2B1_list$celltype,
     HNRNPC_list$celltype, HNRNPM_list$celltype, HNRNPU_list$celltype, ILF2_list$celltype, ILF3_
     list$celltype, NAT10_list$celltype, NONO_list$celltype, RBFOX2_list$celltype, SFPQ_list$
     celltype)

# Generate the plot
ggplot(all, aes(fill=type, y=n, x=name)) +
  geom_bar(position="stack", stat="identity") +
  scale_fill_viridis(discrete = T, labels=c('HepG2_only', '293T_only', "both")) +
  theme_bw() +
  theme(axis.title.x = element_blank(), panel.grid.major = element_blank(), panel.grid.minor =
    element_blank()) + #, legend.position="none") +
  xlab("") +  ylab("# UVC-enrichded proteins")
```




```
# Save the barplot as pdf
pdf("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/Bulk_analysis/2_UVC_enriched/1_Visualization
     /BulkMS_UVC_proteins_stacked.pdf", height = 5, width = 10)
ggplot(all, aes(fill=type, y=n, x=name)) +
  geom_bar(position="stack", stat="identity") +
  scale_fill_viridis(discrete = T) +
  theme_bw() +
  theme(axis.title.x = element_blank(), panel.grid.major = element_blank(), panel.grid.minor =
    element_blank(), legend.position="none") +
  xlab("") +  ylab("# UVC-enrichded proteins")
dev.off()
```

GO analysis of UVC-enriched proteins

```
set.seed(3)
#Run GO analysis on UVC data
gp.res = gost(rownames(UVC_rowData), organism = "hsapiens")

#Take top 20 terms for each source
gp.res <- gp.res$result %>% group_by(source) %>% dplyr::slice(1:10)
gp.bp <- gp.res[gp.res$source %in% c("GO:BP", "GO:CC", "GO:MF"),]
gp.bp$term_name <- factor(gp.bp$term_name, levels = unique(gp.bp$term_name))
gp.bp$source <- factor(gp.bp$source, levels = unique(gp.bp$source))

#Prepare the bubble plot
ggplot(data = gp.bp, aes(x = source, y = term_name, color = -log10(p_value), size = intersection_
    size)) +
  geom_point() +
  scale_color_viridis(option = "D") +
  theme_bw() +
  ylab("") +
  xlab("") +
  theme(axis.text.y = element_text(vjust = 1, hjust=1), axis.text.x = element_text(angle = 90,
    vjust = 0.5, hjust=1))+
  scale_y_discrete(limits=rev)
```




```
# Save the bubble plot as pdf
pdf("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/Bulk_analysis/2_UVC_enriched/1_Visualization
    /GO_UVC_protein.pdf", height = 8, width = 5.5)
ggplot(data = gp.bp, aes(x = source, y = term_name, color = -log10(p_value), size = intersection_
    size)) +
  geom_point() +
  scale_color_viridis(option = "D") +
```

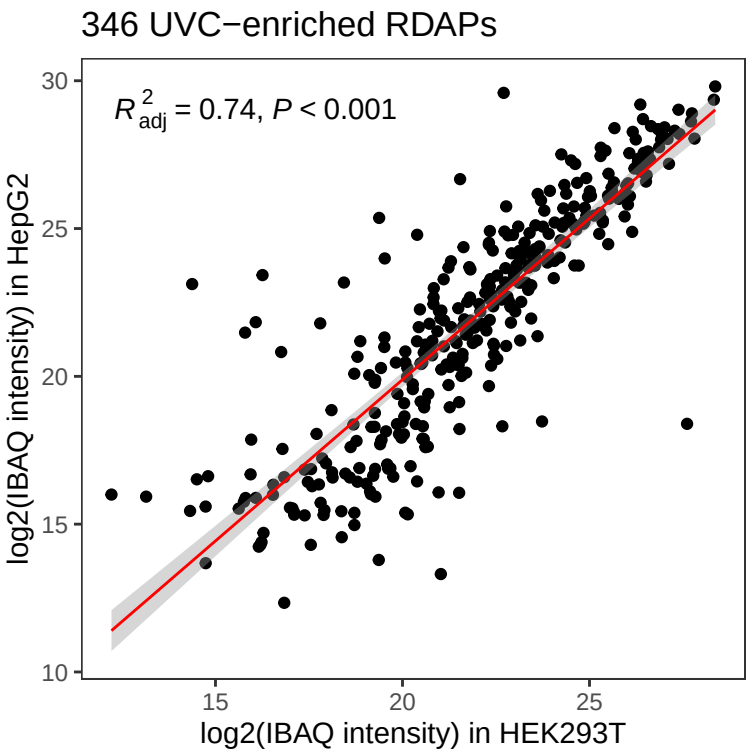
```
  theme_bw() +
  ylab("") +
  xlab("") +
  theme(axis.text.y = element_text(vjust = 1, hjust=1), axis.text.x = element_text(angle = 90,
    vjust = 0.5, hjust=1))+
  scale_y_discrete(limits=rev)
dev.off()
```

Protein expression regression analysis between cell lines

Here, we tested how the UVC-enriched proteins are correlated, in terms of expression, in the two cell lines. To do so, we relied on a previously published dataset (PMID: 22278370).

```
# Load expression from Geiger et al.
data <- read.delim("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/Bulk_analysis/3_Expression_
    profile/0_Reference/Expression_HEK_HepG2.txt")

#Combine the UVC-enriched proteins and the data from Geiger et al.
UVC_anno <- UVC_rowData
UVC_anno$ID <- gsub("\\-.*","",UVC_anno$ID)

data2 <- data %>% mutate(ID = str_split(Uniprot, ";")) %>% unnest(ID)

UVC_exp <- merge(UVC_anno[,515:516], data2[c(3:4,11:23)], by = "ID")

UVC_exp$rowmeans <- rowMeans(UVC_exp[5:10], na.rm = TRUE)
UVC_exp$rowmeans2 <- rowMeans(UVC_exp[11:16], na.rm = TRUE)

UVC_exp2 <- subset(UVC_exp, rowmeans != "NaN")

UVC_exp3 <- UVC_exp2 %>% arrange(ID, -rowmeans) %>% filter(!duplicated(ID))
UVC_exp3 <- UVC_exp3[,c(1:2,5:10)]
UVC_exp3[,3:8] <- 10^UVC_exp3[3:8]

design <- data.frame(label = colnames(UVC_exp3)[3:8], condition = rep(c("H293T", "HepG2"), each =
    3), replicate = rep(c("1", "2", "3"), 2))
columns <- grep("iBAQ", colnames(UVC_exp3))

#Generate a summarizedExperiment
IBAQ.se <- make_se(UVC_exp3, columns = columns, expdesign = design)
IBAQ.se <- filter_se(IBAQ.se, thr = 0)
set.seed(3)
IBAQ.se <- normalize_vsn(IBAQ.se)
IBAQ.se <- DEP2::impute(IBAQ.se, fun = "man", shift = 1.8, scale = 0.3)

UVC_exp4 <- as.data.frame(assay(IBAQ.se))

UVC_exp4$H293T_IBAQ_avg <- rowMeans(UVC_exp4[1:3], na.rm = TRUE)
UVC_exp4$HepG2_IBAQ_avg <- rowMeans(UVC_exp4[4:6], na.rm = TRUE)

h.res.ibaq <- lm(UVC_exp4$H293T_IBAQ_avg ~ UVC_exp4$HepG2_IBAQ_avg)

UVC_exp4$predicted_iBAQ <- predict(h.res.ibaq)
UVC_exp4$residuals_iBAQ <- residuals(h.res.ibaq)

# Plot scatterplot
ggplot(data=UVC_exp4, aes(x=H293T_IBAQ_avg, y=HepG2_IBAQ_avg)) +
  geom_point(color = "black") +
  stat_poly_line(colour = "red", linewidth=0.5) +
  stat_poly_eq(use_label(c("adj.R2", "p.value"))) +
  theme_bw() +
  theme(legend.position = "none", panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
```

```
        panel.background = element_blank(),
        axis.line = element_blank())+
  ggtitle(paste(nrow(UVC_exp4), "UVC-enriched RDAPs")) +
  xlab("log2(IBAQ intensity) in HEK293T") +
  ylab("log2(IBAQ intensity) in HepG2")
```

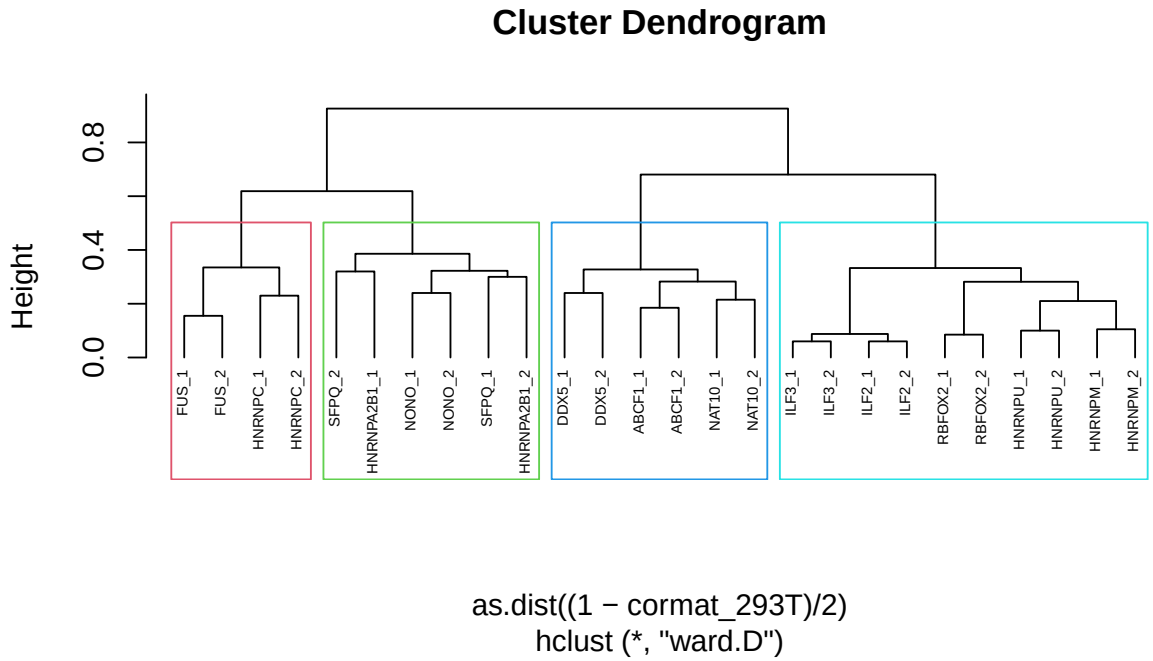



```
UVC_exp4_anno <- as.data.frame(rowData(IBAQ.se))
UVC_exp5 <- merge(UVC_exp4_anno[,1:2], UVC_exp4 , by = "row.names")

write.table(UVC_exp5[,-1], file = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/Bulk_analysis/
    3_Expression_profile/Bulk_MS_Geiger_expression_intensities.txt", row.names = FALSE, sep = "\t
    ")
```

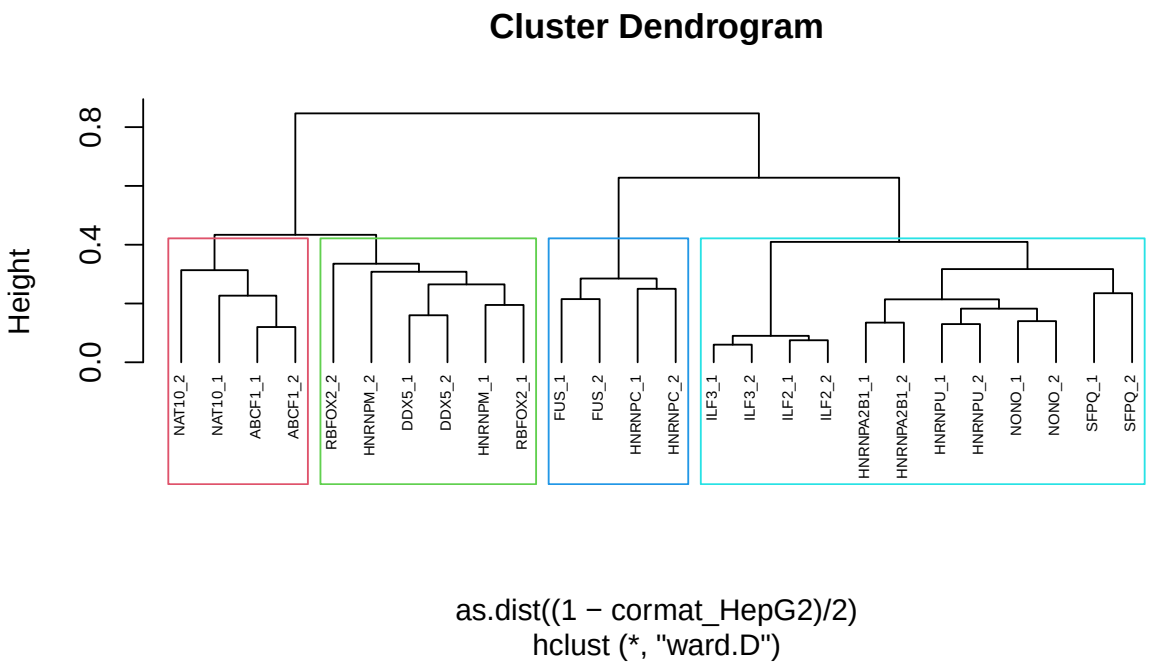
```
# Save scatterplot as pdf
pdf("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/Bulk_analysis/3_Expression_profile/Bulk_MS_
    Geiger_expression.pdf", height = 5, width = 5)
ggplot(data=UVC_exp4, aes(x=H293T_IBAQ_avg, y=HepG2_IBAQ_avg)) +
  geom_point(color = "black") +
  stat_poly_line(colour = "red", linewidth=0.5) +
  stat_poly_eq(use_label(c("adj.R2", "p.value"))) +
  theme_bw() +
  theme(legend.position = "none", panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        axis.line = element_blank())+
  ggtitle(paste(nrow(UVC_exp4), "UVC-enriched RDAPs")) +
  xlab("log2(IBAQ intensity) in HEK293T") +
  ylab("log2(IBAQ intensity) in HepG2")
dev.off()
```

Correlation heatmap of UVC-enriched proteins between HEK293T and HepG2

```
# Generate summarized experiment
set.seed(3)
se_imp <- se_UVC[rownames(UVC_rowData)]
se_imp <- normalize_vsn(se_imp)
se_imp <- DEP2::impute(se_imp, fun = "QRILC")

se_293T_imp <- se_imp[,se_imp$cell_type == "293T" ]
se_HepG2_imp <- se_imp[,se_imp$cell_type == "HepG2"]

# Prepare matrix for clustering
se_293T_log2 <- assay(se_293T_imp)
se_HepG2_log2 <- assay(se_HepG2_imp)

se_293T_log2 <- as.matrix(se_293T_log2)
se_HepG2_log2 <- as.matrix(se_HepG2_log2)

# Calculate the correlation
cormat_293T <- round(cor(se_293T_log2),2)
cormat_HepG2 <- round(cor(se_HepG2_log2),2)
colnames(cormat_293T) <- sub("_UVC_293T_", "_", colnames(cormat_293T))
rownames(cormat_293T) <- sub("_UVC_293T_", "_", rownames(cormat_293T))
colnames(cormat_HepG2) <- sub("_UVC_HepG2_", "_", colnames(cormat_HepG2))
rownames(cormat_HepG2) <- sub("_UVC_HepG2_", "_", rownames(cormat_HepG2))

#Calculate the clustering
hc_293T <- hclust(as.dist((1-cormat_293T)/2), method = "ward.D")
hc_HepG2 <- hclust(as.dist((1-cormat_HepG2)/2), method = "ward.D")

# Plot dendogram
plot(hclust(as.dist((1-cormat_293T)/2), method = "ward.D"), hang=-1, cex=0.5)
rect.hclust(hc_293T, k = 4, border = 2:5)
```

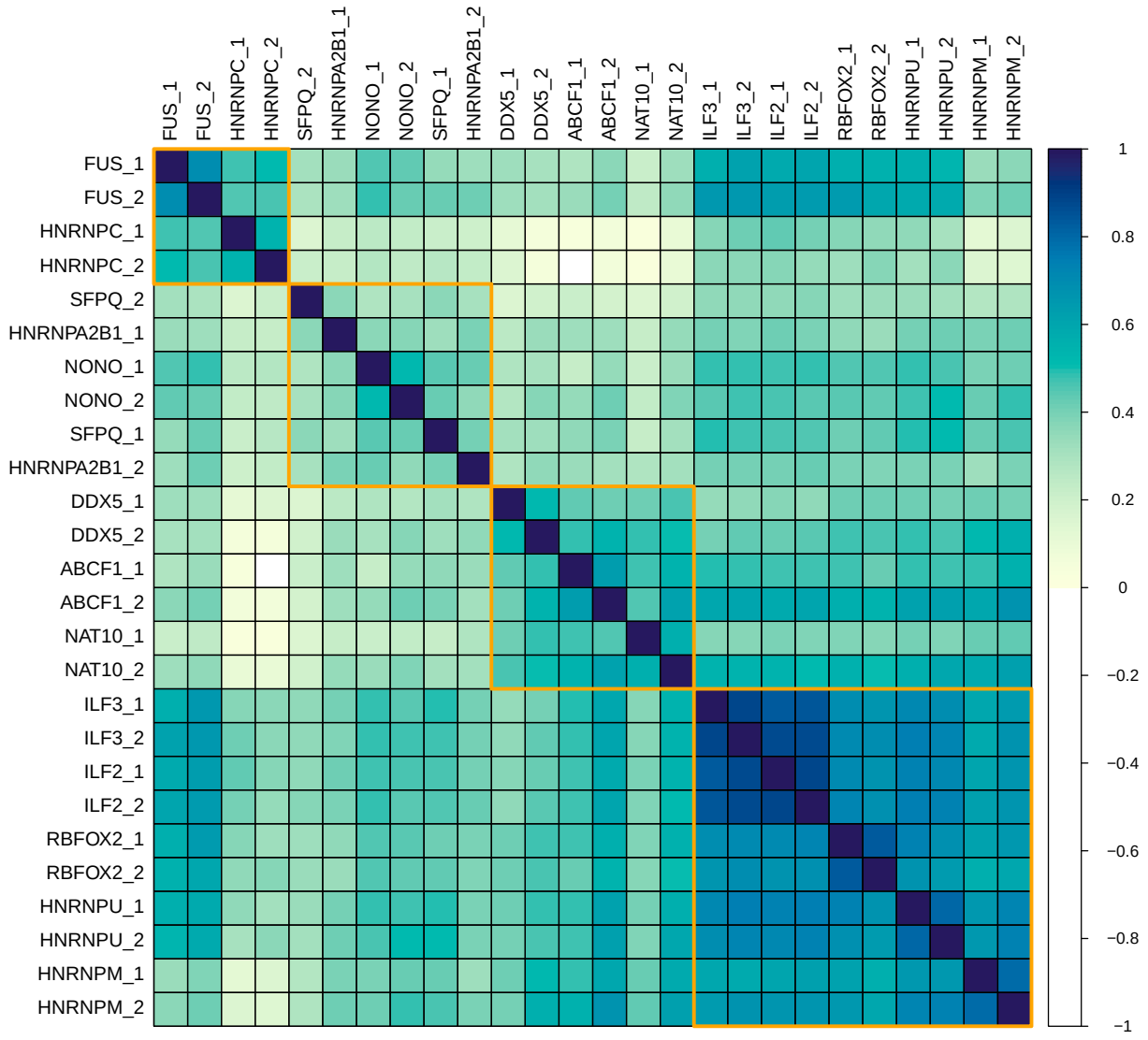



```
plot(hclust(as.dist((1-cormat_HepG2)/2), method = "ward.D"), hang=-1, cex=0.5)
rect.hclust(hc_HepG2, k = 4, border = 2:5)
```

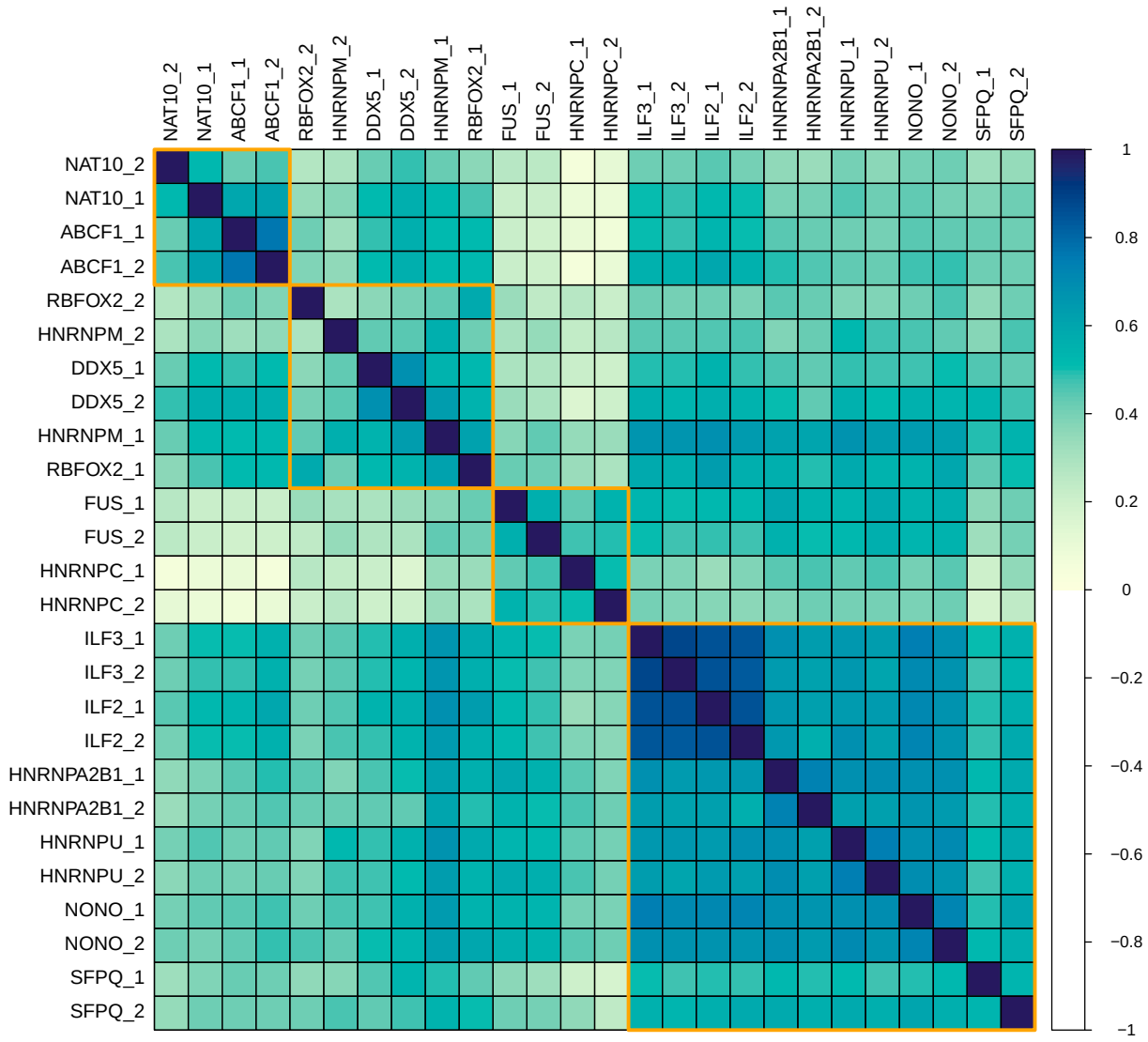



```
# Save dendogram as pdf
pdf("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/Bulk_analysis/4_Correlation/Dendo.pdf")
plot(hclust(as.dist((1-cormat_293T)/2), method = "ward.D"),hang=-1, cex=0.5)
rect.hclust(hc_293T, k = 4, border = 2:5)
plot(hclust(as.dist((1-cormat_HepG2)/2), method = "ward.D"),hang=-1, cex=0.5)
rect.hclust(hc_HepG2, k = 4, border = 2:5)
dev.off()
```

```
#Reorder according to clustering results
d_293T <- as.dendrogram(hc_293T)
ord.293T <- order.dendrogram(d_293T)
cormat.ord_293T <- cormat_293T[ord.293T,ord.293T]

d_HepG2 <- as.dendrogram(hc_HepG2)
ord.HepG2 <- order.dendrogram(d_HepG2)
cormat.ord_HepG2 <- cormat_HepG2[ord.HepG2,ord.HepG2]

#Determine the color
my.breaks <- c(seq(-1, 1, by=0.01))
my.colors <- rev(c(paletteer_c("grDevices::YlGnBu", length(my.breaks)/2), rep("#FFFFFF", length(
    my.breaks)/2)))

#Generate correlogram
corrplot(cormat.ord_293T, method = "color", col = my.colors, number.font = 1, number.cex = 1 ,
    order = "hclust", hclust.method = "ward.D",
        addrect = 4, rect.col = 'orange', rect.lwd = 3, addgrid.col = "black", tl.col = "black",
     diag = TRUE)
```



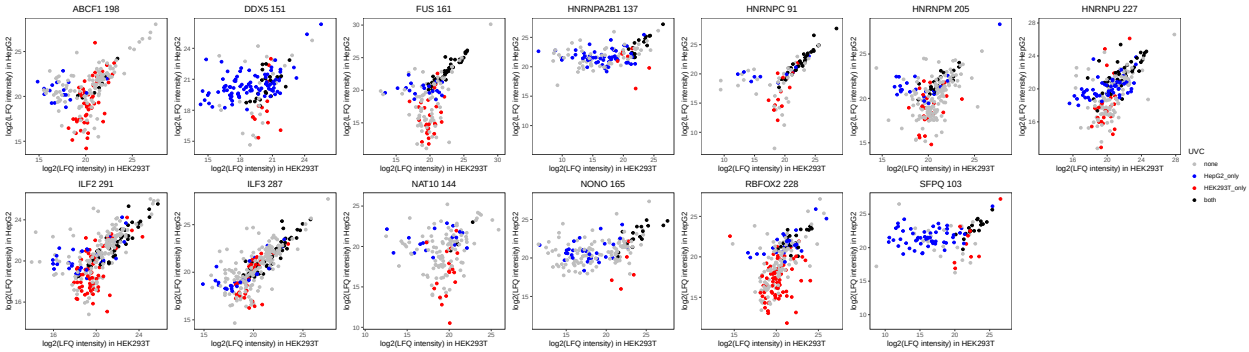

```
corrplot(cormat.ord_HepG2, method = "color", col = my.colors, number.font = 1, number.cex = 1 ,
    order = "hclust", hclust.method = "ward.D",
        addrect = 4, rect.col = 'orange', rect.lwd = 3, addgrid.col = "black", tl.col = "black",
    diag = TRUE)
```




```
# Save the correlation heatmap as pdf
pdf("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/Bulk_analysis/4_Correlation/HM_H293T_HEPG2_
    UVC.pdf", height = 10, width = 10)
corrplot(cormat.ord_293T, method = "color", col = my.colors, number.font = 1, number.cex = 1 ,
    order = "hclust", hclust.method = "ward.D",
        addrect = 4, rect.col = 'orange', rect.lwd = 3, addgrid.col = "black", tl.col = "black",
     diag = TRUE)

corrplot(cormat.ord_HepG2, method = "color", col = my.colors, number.font = 1, number.cex = 1 ,
    order = "hclust", hclust.method = "ward.D",
        addrect = 4, rect.col = 'orange', rect.lwd = 3, addgrid.col = "black", tl.col = "black",
     diag = TRUE)
dev.off()
```

Rank plot of detected proteins

```
#Get the ranking of the proteins
rank_plot_rbp <- function(bait) {
  rbp <- as.data.frame(assay(se_imp[,se_imp$rbp == bait]))
  rbp$name <- rownames(rbp)
  rbp$HEK293T_low_avg <- rowMeans(rbp[,grep("293T", colnames(rbp))])
  rbp$HepG2_low_avg <- rowMeans(rbp[,grep("HepG2", colnames(rbp))])
  rbp <- subset(rbp, name %in% unique(c(rownames(get(paste(bait, "HEK293T_clipper", sep = "_"))$
    all),rownames(get(paste(bait, "HepG2_clipper", sep = "_"))$all))))
  rbp$UVC <- paste(rbp$name %in% rownames(get(paste(bait, "HEK293T_clipper", sep = "_"))$deg),
    rbp$name %in% rownames(get(paste(bait, "HepG2_clipper", sep = "_"))$deg), sep = "_")

  plot <- ggplot(rbp, aes(x=HEK293T_low_avg, y=HepG2_low_avg, label = name)) +
    geom_point(aes(col = UVC)) +
    scale_color_manual(values = c('FALSE_FALSE' = "grey", 'FALSE_TRUE' = "blue", 'TRUE_FALSE' =
    "red", 'TRUE_TRUE' = "black"), labels = c("none", "HepG2_only", "HEK293T_only", "both")) +
    # ggrepel::geom_text_repel(data = subset(ABCF1_rank, Rank < 20), size = 2) +
    theme_bw() +
    theme(legend.position = "none", 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)) +
    xlab("log2(LFQ intensity) in HEK293T") +
    ylab("log2(LFQ intensity) in HepG2") +
    ggtitle(paste(bait, length(rbp$name))) +
    xlim(min(c(rbp$HEK293T_low_avg, rbp$HepG2_low_avg)),max(c(rbp$HEK293T_low_avg, rbp$HepG2_low_
    avg)))+
    ylim(min(c(rbp$HEK293T_low_avg, rbp$HepG2_low_avg)),max(c(rbp$HEK293T_low_avg, rbp$HepG2_low_
    avg)))
  return(plot)
}

ABCF1_plot <- rank_plot_rbp("ABCF1")
DDX5_plot <- rank_plot_rbp("DDX5")
FUS_plot <- rank_plot_rbp("FUS")
HNRNPA2B1_plot <- rank_plot_rbp("HNRNPA2B1")
HNRNPC_plot <- rank_plot_rbp("HNRNPC")
HNRNPM_plot <- rank_plot_rbp("HNRNPM")
HNRNPU_plot <- rank_plot_rbp("HNRNPU")
ILF2_plot <- rank_plot_rbp("ILF2")
ILF3_plot <- rank_plot_rbp("ILF3")
NAT10_plot <- rank_plot_rbp("NAT10")
NONO_plot <- rank_plot_rbp("NONO")
RBFOX2_plot <- rank_plot_rbp("RBFOX2")
SFPQ_plot <- rank_plot_rbp("SFPQ")

# options(ggrepel.max.overlaps = Inf)
ggarrange(ABCF1_plot, DDX5_plot, FUS_plot, HNRNPA2B1_plot, HNRNPC_plot, HNRNPM_plot, HNRNPU_plot,
      ILF2_plot, ILF3_plot, NAT10_plot, NONO_plot, RBFOX2_plot, SFPQ_plot, nrow = 2,  ncol = 7,
    common.legend = TRUE, legend="right")
```

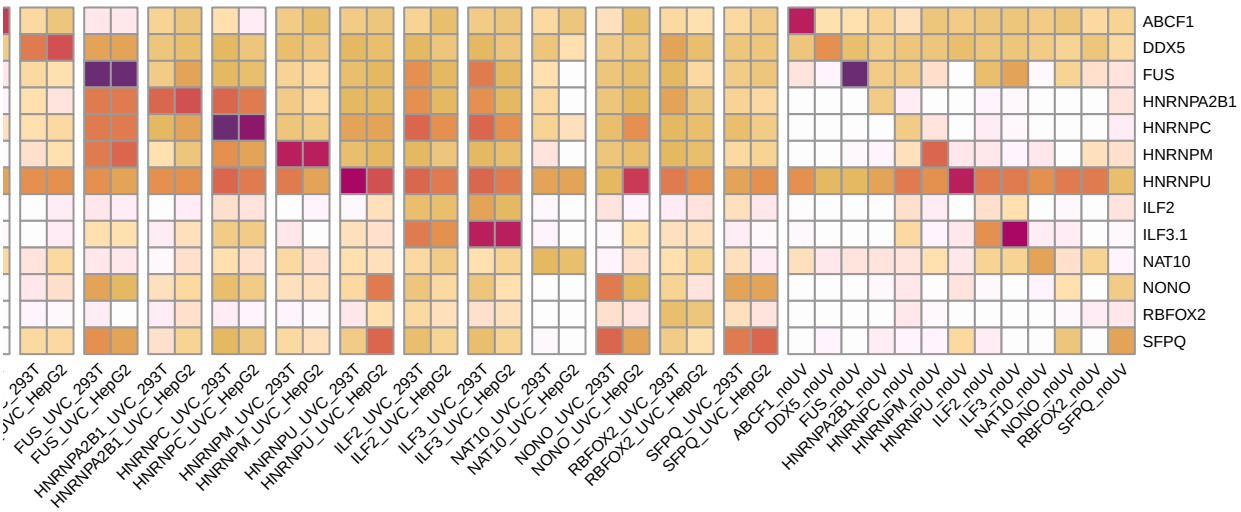



```
#Save the plot as pdf
pdf("~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/Bulk_analysis/2_UVC_enriched/1_Visualization
    /Rank_plot_all.pdf", height = 7, width = 25)
ggarrange(ABCF1_plot, DDX5_plot, FUS_plot, HNRNPA2B1_plot, HNRNPC_plot, HNRNPM_plot, HNRNPU_plot,
      ILF2_plot, ILF3_plot, NAT10_plot, NONO_plot, RBFOX2_plot, SFPQ_plot, nrow = 2,  ncol = 7,
    common.legend = TRUE, legend="right")
dev.off()
```

Reciprocal heatmap of the bait proteins

```
#RBP
rbp <- c("ABCF1", "DDX5", "FUS", "HNRNPA2B1", "HNRNPC", "HNRNPM", "HNRNPU", "ILF2", "ILF3.1", "
    NAT10", "NONO", "RBFOX2", "SFPQ")

#Filter
filt <- se
filt <- filt[rownames(filt) %in% rbp,]
set.seed(3)
se_imp <- DEP2::impute(filt,fun = "QRILC")

df_uvc_imp <- as.data.frame(assay(se_imp))

colnames(df_uvc_imp) <- str_sub(colnames(df_uvc_imp), end = -3)
cn = colnames(df_uvc_imp)
Table.hm <- sapply(unique(cn), function(g) rowMeans(df_uvc_imp[,cn==g,drop=FALSE]))
Table.hm <- Table.hm[order(rownames(Table.hm)),]
Table.hm <- Table.hm[,order(colnames(Table.hm))]

# Color breaks
my.breaks <- c(seq(17, 29, by=0.5))
my.colors <- c(colorRampPalette(colors = rev(c("#6c2c73","#aa0663", "#c93a56", "#db664e", "#
    e4904f", "#e5b961", "#eec67b", "#f7d394", "#ffe0ae", "#ffe0cd", "#ffe7ec", "#fff3fd", "#
    fdfdfd")))(length(my.breaks)))

# Make the heatmap
pheatmap(
  mat               = Table.hm[,c(grep("UVC", colnames(Table.hm)), grep("noUV", colnames(Table.hm
    )))],
  color = my.colors,
  breaks = my.breaks,
  cellwidth = 10,
  cellheight = 10,
  show_colnames     = TRUE,
  show_rownames     = TRUE,
  drop_levels       = TRUE,
  fontsize          = 5.5,
  cluster_rows      = FALSE,
  cluster_cols      = FALSE,
  scale             = "none",
```

```
  angle_col = 45,
  gaps_col = c(2,4,6,8,10,12,14,16,18,20,22,24,26)
)
```




```
pheatmap(
  mat               = Table.hm[,c(grep("UVC", colnames(Table.hm)), grep("noUV", colnames(Table.hm
    )))],
  color = my.colors,
  breaks = my.breaks,
  cellwidth = 10,
  cellheight = 10,
  show_colnames     = TRUE,
  show_rownames     = TRUE,
  drop_levels       = TRUE,
  fontsize          = 5.5,
  cluster_rows      = FALSE,
  cluster_cols      = FALSE,
  scale             = "none",
  angle_col = 45,
  gaps_col = c(2,4,6,8,10,12,14,16,18,20,22,24,26),
  filename = "~/Documents/Postdoc/PD_Projects/3_irCLIP-RNP/MS/Bulk_analysis/4_Correlation/Bait_
    reciprocal_heatmap.pdf",
)
```

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] gprofiler2_0.2.3            ggpmisc_0.5.5
##  [3] ggpp_0.5.6                  psych_2.4.3
##  [5] corrplot_0.92               paletteer_1.6.0
##  [7] factoextra_1.0.7            Clipper_0.0.0.9000
##  [9] UpSetR_1.4.0                ggpubr_0.6.0
## [11] DESeq2_1.38.3               hexbin_1.28.3
## [13] viridis_0.6.5               viridisLite_0.4.2
## [15] ggExtra_0.10.1              textshape_1.7.3
## [17] pacman_0.5.1                hrbrthemes_0.8.7
## [19] gplots_3.1.3.1              RColorBrewer_1.1-3
## [21] pheatmap_1.0.12             data.table_1.15.2
## [23] lubridate_1.9.3             forcats_1.0.0
## [25] stringr_1.5.1               dplyr_1.1.4
## [27] purrr_1.0.2                 readr_2.1.5
## [29] tidyr_1.3.1                 tibble_3.2.1
## [31] ggplot2_3.5.0               tidyverse_2.0.0
## [33] DEP2_0.4.8.24               R6_2.5.1
## [35] limma_3.54.2                MSnbase_2.24.2
## [37] ProtGenerics_1.30.0         mzR_2.32.0
## [39] Rcpp_1.0.12                 MsCoreUtils_1.10.0
## [41] SummarizedExperiment_1.28.0 Biobase_2.58.0
## [43] GenomicRanges_1.50.2        GenomeInfoDb_1.34.9
## [45] IRanges_2.32.0              S4Vectors_0.36.2
## [47] BiocGenerics_0.44.0         MatrixGenerics_1.10.0
## [49] matrixStats_1.2.0
##
## loaded via a namespace (and not attached):
##   [1] SparseM_1.81                missForest_1.5
##   [3] bit64_4.0.5                 knitr_1.45
##   [5] DelayedArray_0.24.0         KEGGREST_1.38.0
##   [7] RCurl_1.98-1.14             AnnotationFilter_1.22.0
##   [9] doParallel_1.0.17           generics_0.1.3
##  [11] preprocessCore_1.60.2       cowplot_1.1.3
##  [13] RSQLite_2.3.5               proxy_0.4-27
##  [15] bit_4.0.5                   tzdb_0.4.0
##  [17] httpuv_1.6.14               assertthat_0.2.1
##  [19] TCseq_1.22.6                xfun_0.42
##  [21] hms_1.1.3                   evaluate_0.23
##  [23] promises_1.2.1              fansi_1.0.6
##  [25] caTools_1.18.2              htmlwidgets_1.6.4
##  [27] igraph_2.0.3                DBI_1.2.2
##  [29] geneplotter_1.76.0          ellipsis_0.3.2
##  [31] RSpectra_0.16-1             QFeatures_1.8.0
##  [33] backports_1.4.1             fontLiberation_0.1.0
##  [35] prismatic_1.1.1             annotate_1.76.0
##  [37] fontBitstreamVera_0.1.1     vctrs_0.6.5
##  [39] imputeLCMD_2.1              quantreg_5.97
##  [41] abind_1.4-5                 cachem_1.0.8
##  [43] withr_3.0.0                 itertools_0.1-3
##  [45] GenomicAlignments_1.34.1    fdrtool_1.2.17
##  [47] MultiAssayExperiment_1.24.0 mnormt_2.1.1
```

```
##  [49] cluster_2.1.6                lazyeval_0.2.2
##  [51] crayon_1.5.2                 crul_1.4.0
##  [53] labeling_0.4.3               glmnet_4.1-8
##  [55] edgeR_3.40.2                 pkgconfig_2.0.3
##  [57] nlme_3.1-164                 rlang_1.1.3
##  [59] lifecycle_1.0.4              miniUI_0.1.1.1
##  [61] sandwich_3.1-0               MatrixModels_0.5-3
##  [63] downloader_0.4               fontquiver_0.2.1
##  [65] httpcode_0.3.0               affyio_1.68.0
##  [67] extrafontdb_1.0              randomForest_4.7-1.1
##  [69] rngtools_1.5.2               Matrix_1.6-5
##  [71] carData_3.0-5                zoo_1.8-12
##  [73] GlobalOptions_0.1.2          png_0.1-8
##  [75] rjson_0.2.21                 bitops_1.0-7
##  [77] KernSmooth_2.23-22           Biostrings_2.66.0
##  [79] blob_1.2.4                   doRNG_1.8.6
##  [81] shape_1.4.6.1                rstatix_0.7.2
##  [83] tmvtnorm_1.6                 ggsignif_0.6.4
##  [85] scales_1.3.0                 memoise_2.0.1
##  [87] magrittr_2.0.3               plyr_1.8.9
##  [89] zlibbioc_1.44.0              compiler_4.2.1
##  [91] pcaMethods_1.90.0            clue_0.3-65
##  [93] Rsamtools_2.14.0             cli_3.6.2
##  [95] affy_1.76.0                  XVector_0.38.0
##  [97] MASS_7.3-60.0.1              tidyselect_1.2.1
##  [99] vsn_3.66.0                   stringi_1.8.3
## [101] highr_0.10                   yaml_2.3.8
## [103] norm_1.0-11.1                askpass_1.2.0
## [105] locfit_1.5-9.9               MALDIquant_1.22.2
## [107] ggrepel_0.9.5                grid_4.2.1
## [109] polynom_1.4-1                tools_4.2.1
## [111] timechange_0.3.0             parallel_4.2.1
## [113] circlize_0.4.16              rstudioapi_0.15.0
## [115] foreach_1.5.2                gridExtra_2.3
## [117] farver_2.1.1                 mzID_1.36.0
## [119] Rtsne_0.17                   digest_0.6.35
## [121] BiocManager_1.30.22          shiny_1.8.0
## [123] gfonts_0.2.0                 car_3.1-2
## [125] broom_1.0.5                  later_1.3.2
## [127] ncdf4_1.22                   httr_1.4.7
## [129] gdtools_0.3.5                AnnotationDbi_1.60.2
## [131] ComplexHeatmap_2.14.0        colorspace_2.1-0
## [133] XML_3.99-0.16.1              reticulate_1.35.0
## [135] umap_0.2.10.0                splines_4.2.1
## [137] rematch2_2.1.2               confintr_1.0.2
## [139] gmm_1.8                      plotly_4.10.4
## [141] systemfonts_1.0.5            xtable_1.8-4
## [143] jsonlite_1.8.8               pillar_1.9.0
## [145] htmltools_0.5.7              mime_0.12
## [147] glue_1.7.0                   fastmap_1.1.1
## [149] BiocParallel_1.32.6          class_7.3-22
## [151] codetools_0.2-19             mvtnorm_1.2-4
## [153] utf8_1.2.4                   lattice_0.22-5
## [155] curl_5.2.1                   gtools_3.9.5
## [157] openssl_2.1.1                Rttf2pt1_1.3.12
## [159] survival_3.5-8               rmarkdown_2.26
## [161] munsell_0.5.0                e1071_1.7-14
## [163] GetoptLong_1.0.5             GenomeInfoDbData_1.2.9
## [165] iterators_1.0.14             impute_1.72.3
## [167] reshape2_1.4.4               gtable_0.3.4
## [169] extrafont_0.19
```