

Figure4 RAPseq

Qun Li

2 Sep 2024 (10:40:08)

Contents

0.1	ENV settings	1
0.2	R libraries	2
0.3	Color settings	7
0.4	Load data	7
0.5	Figures	7
0.5.1	Figure 4A	7
0.5.2	Figure 4B	10
0.5.3	Figure 4E	14
0.5.4	Figure 4F	18
0.5.5	Figure S4A	20
0.5.6	Figure S4B	22
0.5.7	Figure S4C	24
0.5.8	Figure S4D	26
0.5.9	Figure S4E right	29
0.5.10	Figure S4F	31
0.5.11	Figure S4G	33

0.1 ENV settings

```
###  
# @Description: Figure4  
# @Description: Adapted from https://github.com/IonutAtanasoai1/RAPseq  
# @Author: LiQun  
# @Email: qun.li@ki.se  
# @Date: 2 Sep 2024 ( 10:40:08 )  
###  
rm(list = ls())  
setwd("/Users/liqun/Desktop/Projects/RAPseq/AnalysisQun/")
```

0.2 R libraries

```
library(ChIPpeakAnno)
## Loading required package: IRanges
## Loading required package: BiocGenerics
##
## Attaching package: 'BiocGenerics'
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which.max, which.min
## Loading required package: S4Vectors
## Loading required package: stats4
##
## Attaching package: 'S4Vectors'
## The following object is masked from 'package:utils':
##
##     findMatches
## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname
## Loading required package: GenomicRanges
## Loading required package: GenomeInfoDb
## Warning: package 'GenomeInfoDb' was built under R version 4.3.3
library(edgeR)
## Loading required package: limma
##
## Attaching package: 'limma'
## The following object is masked from 'package:BiocGenerics':
##
##     plotMA
library(ggfortify)
## Loading required package: ggplot2
library(tidyverse)
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v lubridate  1.9.3      v tibble    3.2.1
## v purrr      1.0.2      v tidyr     1.3.1
## -- Conflicts ----- tidyverse_conflicts() --
## x lubridate::%within%() masks IRanges::%within%()
## x dplyr::collapse()     masks IRanges::collapse()
## x dplyr::combine()     masks BiocGenerics::combine()
## x dplyr::desc()        masks IRanges::desc()
## x tidyr::expand()       masks S4Vectors::expand()
## x dplyr::filter()       masks stats::filter()
```

```

## x dplyr::first()      masks S4Vectors::first()
## x dplyr::lag()        masks stats::lag()
## x ggplot2::Position() masks BiocGenerics::Position(), base::Position()
## x purrr::reduce()     masks GenomicRanges::reduce(), IRanges::reduce()
## x dplyr::rename()     masks S4Vectors::rename()
## x lubridate::second() masks S4Vectors::second()
## x lubridate::second<-() masks S4Vectors::second<-()
## x dplyr::slice()      masks IRanges::slice()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
library(scales)
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##   discard
##
## The following object is masked from 'package:readr':
##
##   col_factor
library(ggplot2)
library(dendextend)
##
## -----
## Welcome to dendextend version 1.17.1
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
## To suppress this message use: suppressPackageStartupMessages(library(dendextend))
## -----
##
##
## Attaching package: 'dendextend'
##
## The following object is masked from 'package:stats':
##
##   cutree
library(dichromat)
library(reshape2)
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##   smiths
library(dplyr)
library(stringr)

```

```

library(GenomicRanges)
library(GenomicFeatures)
## Warning: package 'GenomicFeatures' was built under R version 4.3.3
## Loading required package: AnnotationDbi
## Loading required package: Biobase
## Welcome to Bioconductor
##
##      Vignettes contain introductory material; view with
##      'browseVignettes()'. To cite Bioconductor, see
##      'citation("Biobase")', and for packages 'citation("pkgname")'.
##
##
## Attaching package: 'AnnotationDbi'
##
## The following object is masked from 'package:dplyr':
##
##      select
library(gplots)
##
## Attaching package: 'gplots'
##
## The following object is masked from 'package:IRanges':
##
##      space
##
## The following object is masked from 'package:S4Vectors':
##
##      space
##
## The following object is masked from 'package:stats':
##
##      lowess
library(clusterProfiler)
## Warning: package 'clusterProfiler' was built under R version 4.3.3
##
## clusterProfiler v4.10.1 For help: https://yulab-smu.top/biomedical-knowledge-mining-book/
##
## If you use clusterProfiler in published research, please cite:
## T Wu, E Hu, S Xu, M Chen, P Guo, Z Dai, T Feng, L Zhou, W Tang, L Zhan, X Fu, S Liu, X Bo, and G Yu.
##
## Attaching package: 'clusterProfiler'
##
## The following object is masked from 'package:AnnotationDbi':
##
##      select
##
## The following object is masked from 'package:purrr':
##
##      simplify
##
## The following object is masked from 'package:IRanges':
##
##      slice

```

```

##
## The following object is masked from 'package:S4Vectors':
##
##     rename
##
## The following object is masked from 'package:stats':
##
##     filter
library(org.Hs.eg.db)
##
library(org.Dr.eg.db)
##
library(tidyr)
library(gtools)
library(ggrepel)
library(ggbeeswarm)
library(ReactomePA)
## ReactomePA v1.46.0 For help: https://yulab-smu.top/biomedical-knowledge-mining-book/
##
## If you use ReactomePA in published research, please cite:
## Guangchuang Yu, Qing-Yu He. ReactomePA: an R/Bioconductor package for reactome pathway analysis and
library(VennDiagram)
## Loading required package: grid
## Loading required package: futile.logger
##
## Attaching package: 'futile.logger'
##
## The following object is masked from 'package:gtools':
##
##     scat
##
## Attaching package: 'VennDiagram'
##
## The following object is masked from 'package:dendextend':
##
##     rotate
library(LSD)
library(gridExtra)
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:Biobase':
##
##     combine
##
## The following object is masked from 'package:dplyr':
##
##     combine
##
## The following object is masked from 'package:BiocGenerics':
##
##     combine

```

```

library(UpSetR)
library(ggpubr)
##
## Attaching package: 'ggpubr'
##
## The following object is masked from 'package:VennDiagram':
##
##     rotate
##
## The following object is masked from 'package:dendextend':
##
##     rotate
library(GenomicScores)
##
## Attaching package: 'GenomicScores'
##
## The following object is masked from 'package:utils':
##
##     citation
library(phastCons7way.UCSC.hg38)
## Warning: replacing previous import 'utils::findMatches' by
## 'S4Vectors::findMatches' when loading 'phastCons7way.UCSC.hg38'
library(phastCons100way.UCSC.hg38)
## Warning: replacing previous import 'utils::findMatches' by
## 'S4Vectors::findMatches' when loading 'phastCons100way.UCSC.hg38'
library(corrplot)
## Warning: package 'corrplot' was built under R version 4.3.3
## corrplot 0.94 loaded
library(ggforce)
library(idr)
library(GGally)
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg      ggplot2
library(eulerr)
## Registered S3 method overwritten by 'eulerr':
##   method from
##   plot.venn gplots
##
## Attaching package: 'eulerr'
##
## The following object is masked from 'package:gplots':
##
##     venn
library(ggseqlogo)
library(robustbase)
## Warning: package 'robustbase' was built under R version 4.3.3
##
## Attaching package: 'robustbase'
##
## The following object is masked from 'package:Biobase':
##
##     rowMedians

```

```
library(rrvgo)
## Warning: package 'rrvgo' was built under R version 4.3.3
```

0.3 Color settings

```
acqua_greens <- c("#04493F", "#02655B", "#08756A", "#0C8074", "#179286",
                  "#25A296", "#3EB3A7", "#5FCBC1", "#84E2D9", "#A4F2EA",
                  "#C4FCF6", "#E2FFFD")
acqua_blues <- c("#02465B", "#015666", "#0C6476", "#157589", "#1E8498",
                 "#4397A8", "#4FB0C3", "#70C2D2", "#8DDDBEB", "#C1EDF6",
                 "#D7F8FF")
greys <- c("#202020", "#404040", "#606060", "#808080", "#A0A0A0", "#C0C0C0",
           "#E0E0E0", "#FFFFFF")
reds <- c("#C10303", "#D83535", "#E95F5F", "#F08686", "#FAAEAE")
pinks <- c("#660000", "#990000", "#CC0000", "#FF0000", "#FF6666", "#FF9999",
           "#FFCECE")
yellows <- c("#9C8803", "#A59213", "#B09D1F", "#BAA82F", "#C6B441", "#D0BF53",
             "#DACB69", "#E4D782", "#EEE39D", "#FAF2BA", "#FFFBD1")
oranges <- c("#A15000", "#CE7012", "#E47B12", "#F07D09", "#FF8B15", "#FFA141",
             "#FFCC99")
```

0.4 Load data

```
# RAP data
HUR <- read.table("./Data/RAPseq/annotated/hsHuRHUMAN.peaks.txt",
                  stringsAsFactors = F, header = T)
PTBP1 <- read.table("./Data/RAPseq/annotated/PTBP1.peaks.txt",
                    stringsAsFactors = F, header = T)
HUR_PTBP1 <- read.table("./Data/RAPseq/annotated/HuRPTBP1.peaks.txt",
                        stringsAsFactors = F, header = T)

print(nrow(HUR))
## [1] 8073
print(nrow(PTBP1))
## [1] 12176
print(nrow(HUR_PTBP1))
## [1] 15029
```

0.5 Figures

0.5.1 Figure 4A

```
# HUR
HUR_selected_forPlot <- HUR[,c(1,7,8,4:6)]
HUR_selected_forPlot$Summit_start <- HUR_selected_forPlot$Summit_start - 50
HUR_selected_forPlot$Summit_end <- HUR_selected_forPlot$Summit_end + 50
# PTBP1
PTBP1_selected_forPlot <- PTBP1[,c(1,7,8,4:6)]
```

```

PTBP1_selected_forPlot$Summit_start <- PTBP1_selected_forPlot$Summit_start - 50
PTBP1_selected_forPlot$Summit_end <- PTBP1_selected_forPlot$Summit_end + 50

HUR_GRanges <- makeGRangesFromDataFrame(HUR_selected_forPlot)
PTBP1_GRanges <- makeGRangesFromDataFrame(PTBP1_selected_forPlot)

# uniq and common
HUR_GRanges_uniq <- unique(HUR[-as.data.frame(findOverlaps(HUR_GRanges,PTBP1_GRanges,type = "any"))[,1],
HUR_PTBP1_GRanges_common <- as.character(seq(1,nrow( as.data.frame(findOverlaps(HUR_GRanges,PTBP1_GRanges
HUR_GRanges_merge <- c(HUR_GRanges_uniq,HUR_PTBP1_GRanges_common)
PTBP1_GRanges_uniq <- unique(PTBP1[-as.data.frame(findOverlaps(HUR_GRanges,PTBP1_GRanges,type = "any"))
PTBP1_GRanges_merge <- c(PTBP1_GRanges_uniq,HUR_PTBP1_GRanges_common)

HUR_PTBP1_GRanges_list <- list(HUR_GRanges_merge, PTBP1_GRanges_merge)

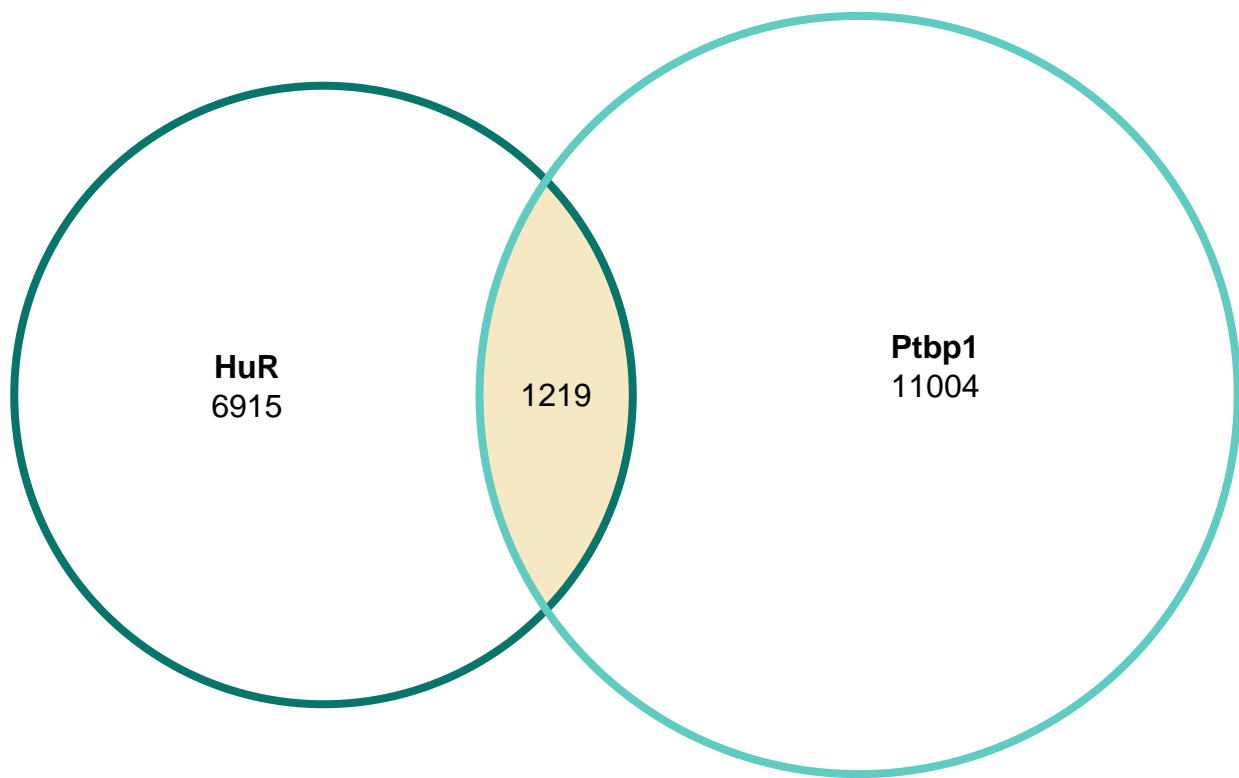
names(HUR_PTBP1_GRanges_list) <- c("HuR", "Ptbp1")

# plot
HUR_PTBP1_overlaps_forPlot <- euler(HUR_PTBP1_GRanges_list, shape="circle")

HUR_overlaps <- HUR[as.data.frame(
  findOverlaps(HUR_GRanges, PTBP1_GRanges, type = "any"))[,1],
  "Summit_start"]
PTBP1_overlaps <- PTBP1[as.data.frame(
  findOverlaps(HUR_GRanges,PTBP1_GRanges, type = "any"))[,2],
  "Summit_start"]

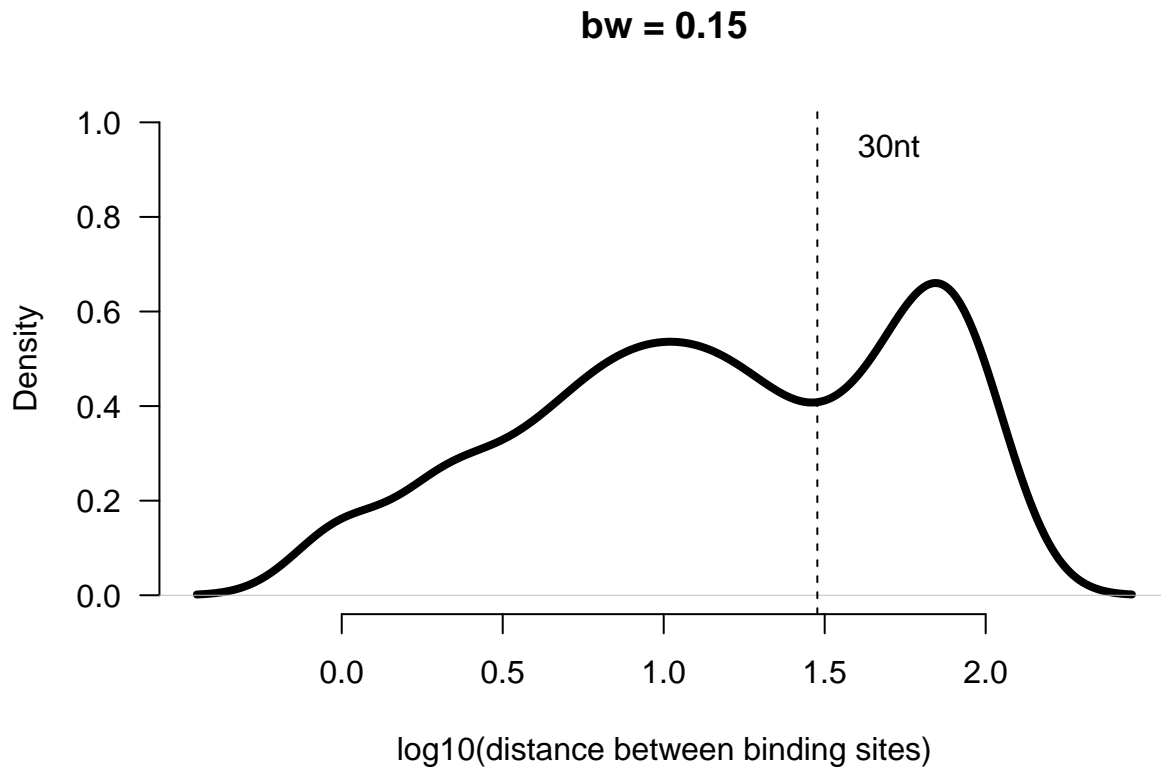
#pdf("./Figure/Figure4/Figure4A_1.pdf", width = 8, height = 5)
plot(HUR_PTBP1_overlaps_forPlot, fills=c("white", "white", "#f6e8c3"),
     quantities=TRUE, edges=T, col=c(acqua_greens[c(3,8)]), lwd=4)

```

```
#dev.off()

#pdf("./Figure/Figure4/Figure4A_2.pdf", width = 8, height = 5)
par(bty="n")
plot(density(log10(abs(PTBP1_overlaps-HUR_overlaps))), bw=0.15),
      xlab="log10(distance between binding sites)", main="bw = 0.15",
      ylim=c(0,1), las=1, lwd=4, xaxt="n")
axis(1,at=c(0,0.5,1,1.5,2))
abline(v=log10(30), lty=2)
text(x=1.7,y=0.95,label="30nt")
```



```
#dev.off()
```

0.5.2 Figure 4B

```
# extract 2 replicates
HUR_replicates <- HUR[as.data.frame(findOverlaps(HUR_GRanges, PTBP1_GRanges, type = "any"))[,1],][c("R", "L"),]
PTBP1_replicates <- PTBP1[as.data.frame(findOverlaps(HUR_GRanges, PTBP1_GRanges, type = "any"))[,2],][c("R", "L"),]
HUR_comp <- HUR_replicates[log10(abs(PTBP1_overlaps-HUR_overlaps)+1)<1.492,]
PTBP1_comp <- PTBP1_replicates[log10(abs(PTBP1_overlaps-HUR_overlaps)+1)<1.492,]

HUR_rep1 <- HUR_comp$Rep1
HUR_rep2 <- HUR_comp$Rep2
PTBP1_rep1 <- PTBP1_comp$Rep1
PTBP1_rep2 <- PTBP1_comp$Rep2

# merge two data
HUR_PTBP1_tworeplicates <- data.frame(HUR_rep1, HUR_rep2, PTBP1_rep1, PTBP1_rep2)

# DEG analysis
group <- c(1,1,2,2)
DEG_data <- DGEList(counts=HUR_PTBP1_tworeplicates, group = group)
design <- model.matrix(~group)
DEG_data <- estimateDisp(DEG_data,design)
```

```

fit <- glmQLFit(DEG_data, design)
qlf <- glmQLFTest(fit)

DEG_data_table <- qlf$table
DEG_data_table$p.adjust <- -log10(p.adjust(DEG_data_table$PValue, method = "BH"))
DEG_data_table$PValue <- -log10(DEG_data_table$PValue)

# HUR_PTBP1_tworeplicates
HUR_PTBP1_tworeplicates$HuR <- log2((HUR_PTBP1_tworeplicates$HUR_rep1 + HUR_PTBP1_tworeplicates$HUR_rep2)/2)
HUR_PTBP1_tworeplicates$PTBP1 <- log2((HUR_PTBP1_tworeplicates$PTBP1_rep1 + HUR_PTBP1_tworeplicates$PTBP1_rep2)/2)
HUR_PTBP1_tworeplicates$delta_BS <- abs(HUR_PTBP1_tworeplicates$HuR - HUR_PTBP1_tworeplicates$PTBP1)
HUR_PTBP1_tworeplicates$FDR <- DEG_data_table$p.adjust
HUR_PTBP1_tworeplicates$Significant <- HUR_PTBP1_tworeplicates$FDR >= 1.30103
HUR_PTBP1_tworeplicates$logFC <- DEG_data_table$logFC

HUR_PTBP1_tworeplicates$HuR_won <- HUR_PTBP1_tworeplicates$logFC <= -1 & HUR_PTBP1_tworeplicates$FDR >= 0.05
HUR_PTBP1_tworeplicates$PTBP1_won <- HUR_PTBP1_tworeplicates$logFC >= 1 & HUR_PTBP1_tworeplicates$FDR >= 0.05
HUR_PTBP1_tworeplicates$DiffBind <- paste(HUR_PTBP1_tworeplicates$HuR_won, HUR_PTBP1_tworeplicates$PTBP1_won)
HUR_PTBP1_tworeplicates$DiffBind <- gsub("FALSE_FALSE", "N.S.", HUR_PTBP1_tworeplicates$DiffBind)
HUR_PTBP1_tworeplicates$DiffBind <- gsub("FALSE_TRUE", "PTBP1", HUR_PTBP1_tworeplicates$DiffBind)
HUR_PTBP1_tworeplicates$DiffBind <- gsub("TRUE_FALSE", "HUR", HUR_PTBP1_tworeplicates$DiffBind)

#pdf("./Figure/Figure4/Figure4B.pdf", width = 8, height = 5)

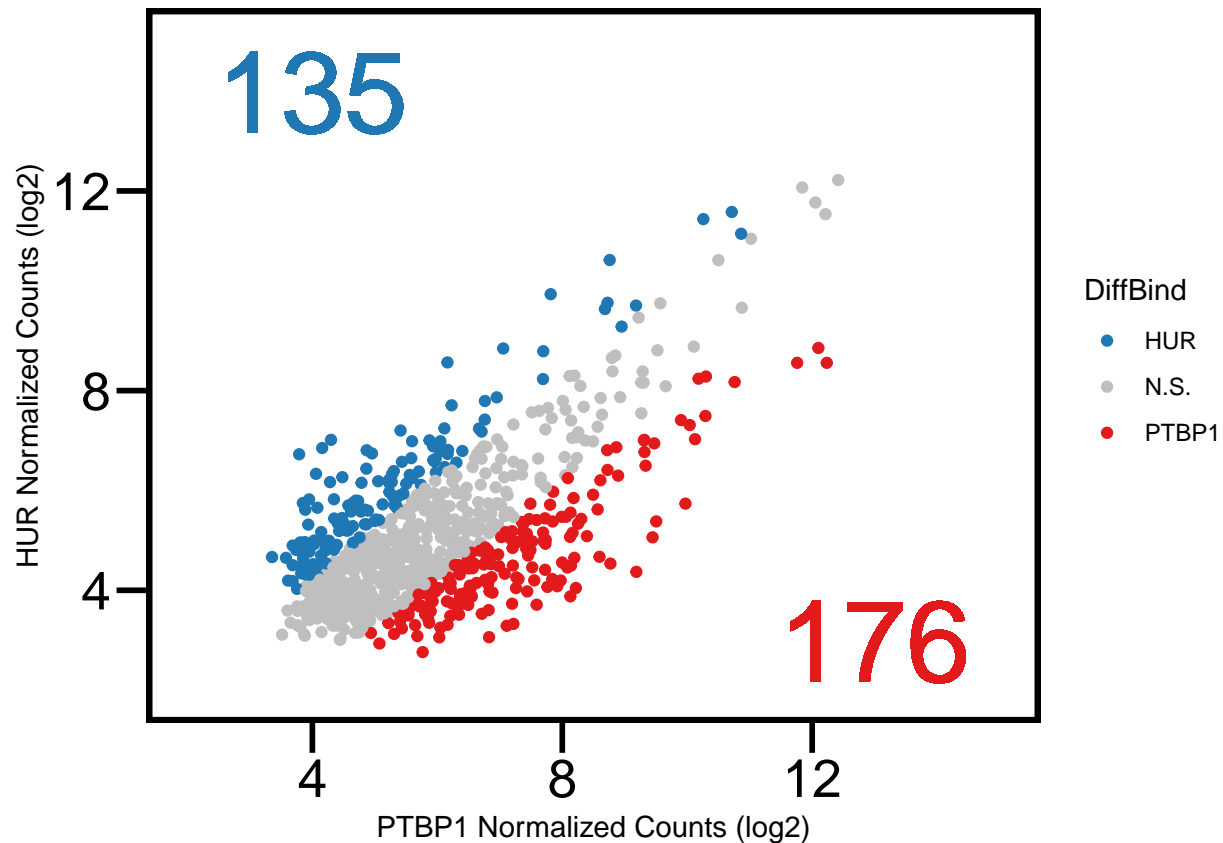
HUR_PTBP1_DEGPlot <- ggplot(data=HUR_PTBP1_tworeplicates) +
  geom_point(aes(x=PTBP1, y=HuR, color=DiffBind)) +
  scale_color_manual(values = c("#1f78b4", "grey75", "#e31a1c")) +
  theme(panel.background = element_blank(),
        panel.grid = element_blank(),
        panel.border = element_rect(size=2, fill=NA, color="black"),
        axis.ticks.length = unit(4, "mm"),
        axis.ticks = element_line(color="black", size=1),
        axis.text = element_text(color="black", size=20)) +
  xlim(2,15) +
  ylim(2,15) +
  xlab("PTBP1 Normalized Counts (log2)") +
  ylab("HUR Normalized Counts (log2)")

## Warning: The `size` argument of `element_line()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
## Warning: The `size` argument of `element_rect()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

HUR_PTBP1_DEGPlot + geom_text(aes(x=13, y=3), label = table(HUR_PTBP1_tworeplicates$DiffBind)["PTBP1"], size=15, color="black") +
  geom_text(aes(x=4, y=14), label = table(HUR_PTBP1_tworeplicates$DiffBind)["HUR"], size=15, color="#1f78b4")
## Warning in geom_text(aes(x = 13, y = 3), label = table(HUR_PTBP1_tworeplicates$DiffBind)["PTBP1"], size=15, color="black"):
## i Please consider using `annotate()` or provide this layer with data containing
## a single row.

```

```
## Warning in geom_text(aes(x = 4, y = 14), label = table(HUR_PTBP1_tworeplicates$DiffBind)["HUR"], : A
## i Please consider using `annotate()` or provide this layer with data containing
## a single row.
```



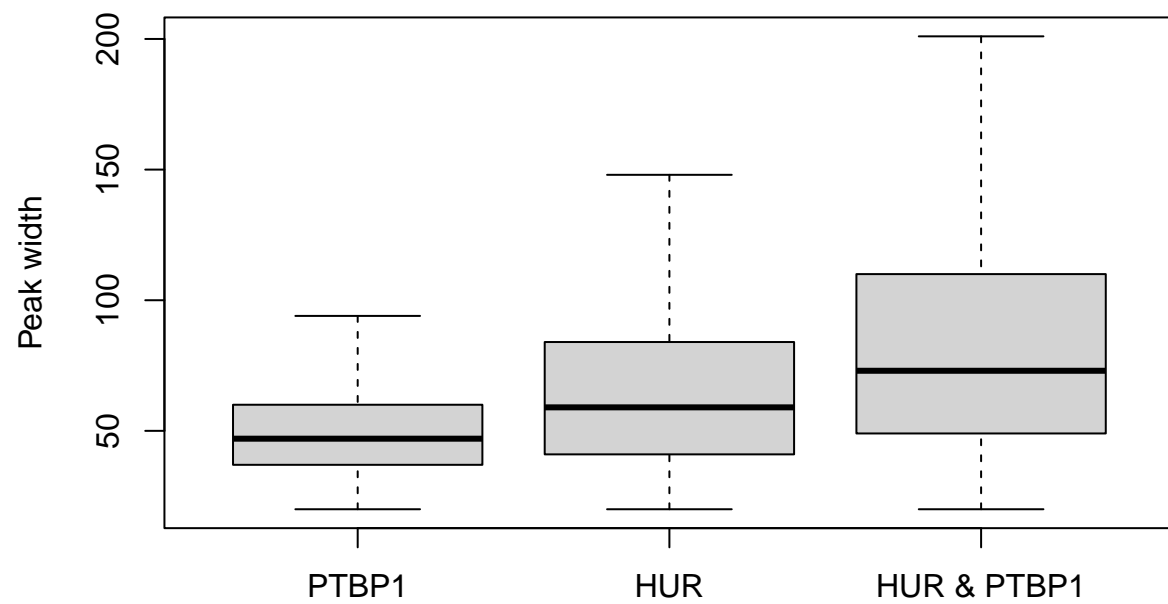
```
#dev.off()

# median sequenced fragment size HUR = 50
# median sequenced fragment size HURPTBP1_coRAP = 51
# median sequenced fragment size PTBP1 = 35

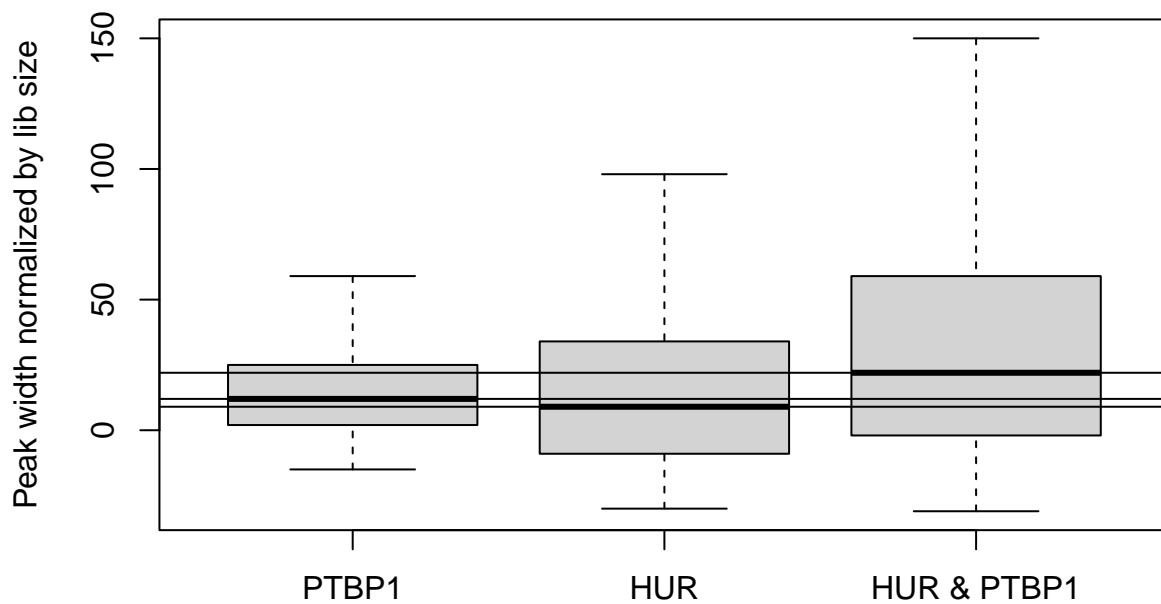
PTBP1_len <- PTBP1$end - PTBP1$start
PTBP1_len_adjust <- PTBP1_len - 35
HUR_len <- HUR$end - HUR$start
HUR_len_adjust <- HUR_len - 50
HUR_PTBP1_len <- HUR_PTBP1$end - HUR_PTBP1$start
HUR_PTBP1_len_adjust <- HUR_PTBP1_len - 51

HUR_PTBP1_len_list <- list(PTBP1_len, HUR_len, HUR_PTBP1_len)
names(HUR_PTBP1_len_list) <- c("PTBP1", "HUR", "HUR & PTBP1")
HUR_PTBP1_len_adjust_list <- list(PTBP1_len_adjust, HUR_len_adjust, HUR_PTBP1_len_adjust)
names(HUR_PTBP1_len_adjust_list) <- c("PTBP1", "HUR", "HUR & PTBP1")

boxplot(HUR_PTBP1_len_list, outline=F, ylab="Peak width")
```



```
boxplot(HUR_PTBP1_len_adjust_list, outline=F, ylab="Peak width normalized by lib size")
abline(h=c(median(PTBP1_len_adjust), median(HUR_len_adjust), median(HUR_PTBP1_len_adjust)))
```



0.5.3 Figure 4E

```
# median sequenced fragment size HUR = 50
# median sequenced fragment size HURPTBP1 = 51
# median sequenced fragment size PTBP1 = 35

# HUR
HUR_selected_forPlot <- HUR[,c(1,7,8,4:6)]
HUR_selected_forPlot$Summit_start <- HUR_selected_forPlot$Summit_start - 50
HUR_selected_forPlot$Summit_end <- HUR_selected_forPlot$Summit_end + 50

# PTBP1
PTBP1_selected_forPlot <- PTBP1[,c(1,7,8,4:6)]
PTBP1_selected_forPlot$Summit_start <- PTBP1_selected_forPlot$Summit_start - 50
PTBP1_selected_forPlot$Summit_end <- PTBP1_selected_forPlot$Summit_end + 50

# HUR_PTBP1
HUR_PTBP1_selected_forPlot <- HUR_PTBP1[,c(1,7,8,4:6)]
HUR_PTBP1_selected_forPlot$Summit_start <- HUR_PTBP1_selected_forPlot$Summit_start - 50
HUR_PTBP1_selected_forPlot$Summit_end <- HUR_PTBP1_selected_forPlot$Summit_end + 50

HUR_GRanges <- makeGRangesFromDataFrame(HUR_selected_forPlot)
PTBP1_GRanges <- makeGRangesFromDataFrame(PTBP1_selected_forPlot)
HUR_PTBP1_GRanges <- makeGRangesFromDataFrame(HUR_PTBP1_selected_forPlot)
```

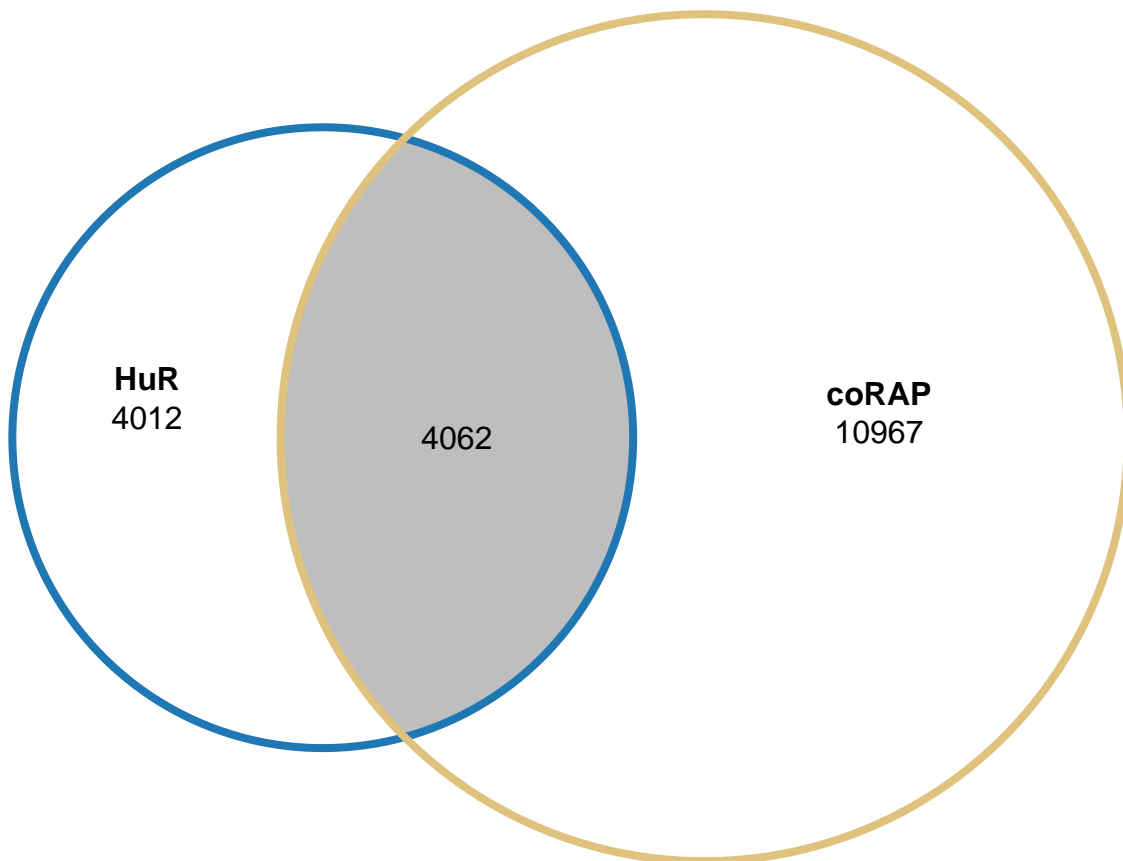
```

# uniq and common
HUR_GRanges_uniq <- unique(HUR[-as.data.frame(findOverlaps(HUR_GRanges,HUR_PTBP1_GRanges,type = "any"))
HUR_PTBP1_GRanges_uniq <- unique(HUR_PTBP1[as.data.frame(findOverlaps(HUR_GRanges,HUR_PTBP1_GRanges,type
HUR_PTBP1_GRanges_merge <- c(HUR_GRanges_uniq,HUR_PTBP1_GRanges_uniq)
coRAP1 <- unique(HUR_PTBP1[-as.data.frame(findOverlaps(HUR_GRanges,HUR_PTBP1_GRanges,type = "any"))[,2]
coRAP1 <- c(coRAP1,HUR_PTBP1_GRanges_uniq)
HUR_coRAP <- list(HUR_PTBP1_GRanges_merge, coRAP1)
names(HUR_coRAP) <- c("HuR", "coRAP")
HUR_coRAP_overlaps_forPlot <- euler(HUR_coRAP, shape="circle")

PTBP1_GRanges_uniq <- unique(PTBP1[-as.data.frame(findOverlaps(PTBP1_GRanges,HUR_PTBP1_GRanges,type = "
PTBP1_HUR_GRanges_uniq <- unique(HUR_PTBP1[as.data.frame(findOverlaps(PTBP1_GRanges,HUR_PTBP1_GRanges,t
PTBP1_HUR_GRanges_merge <- c(PTBP1_GRanges_uniq,PTBP1_HUR_GRanges_uniq)
coRAP2 <- unique(HUR_PTBP1[-as.data.frame(findOverlaps(PTBP1_GRanges,HUR_PTBP1_GRanges,type = "any"))[,2]
coRAP2 <- c(coRAP2,PTBP1_HUR_GRanges_uniq)
PTBP1_coRAP <- list(PTBP1_HUR_GRanges_merge, coRAP2)
names(PTBP1_coRAP) <- c("PTBP1", "coRAP")
PTBP1_coRAP_overlaps_forPlot <- euler(PTBP1_coRAP, shape="circle")

# plot
#pdf("./Figure/Figure4/Figure4E_Venn_HuR.pdf", width = 6, height = 3)
plot(HUR_coRAP_overlaps_forPlot, fills=c("white", "white", "grey"), quantities=TRUE, edges=T, col=c("#1

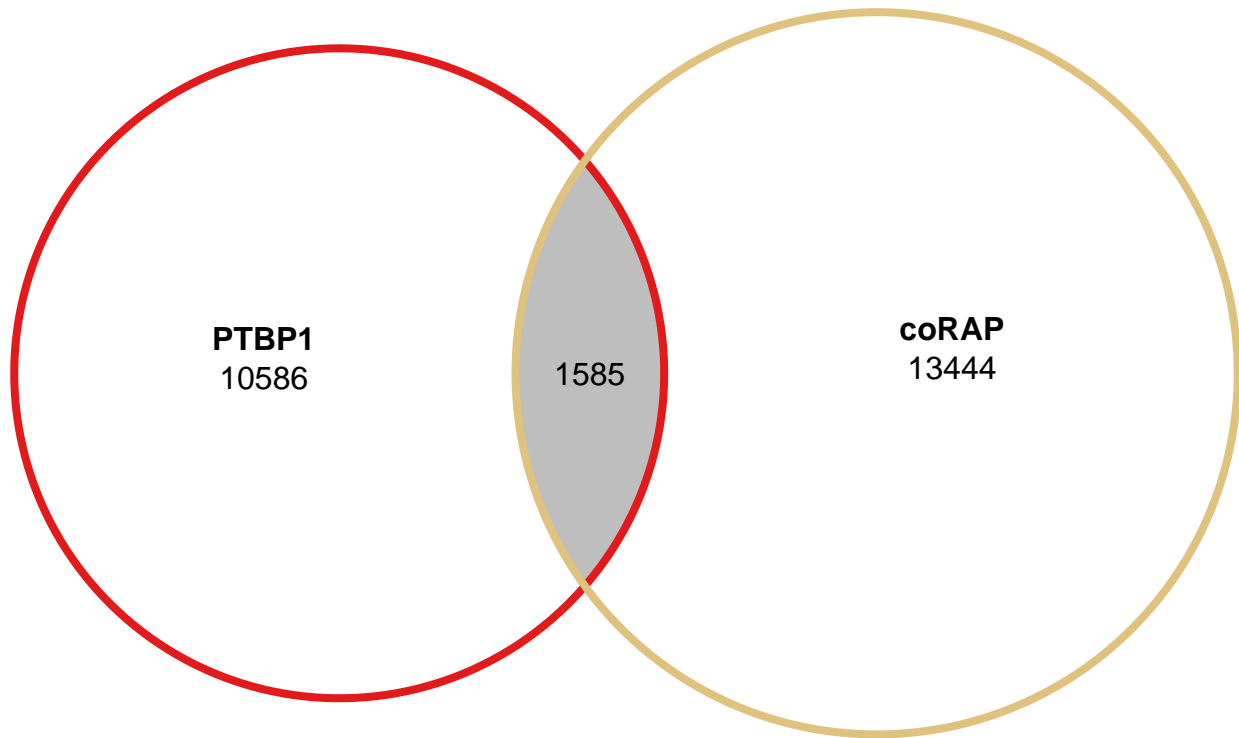
```



```
#dev.off()
```

```
#pdf("./Figure/Figure4/Figure4E_Venn_PTBP1.pdf", width = 6, height = 3)
```

```
plot(PTBP1_coRAP_overlaps_forPlot, fills=c("white", "white", "grey"), quantities=TRUE, edges=T, col=c("red", "yellow", "grey"))
```



```
#dev.off()
```

```
# overlaps
```

```
HUR_PTBP1_HURPTBP1_overlaps <- HUR_PTBP1[as.data.frame(findOverlaps(HUR_GRanges,HUR_PTBP1_GRanges,type="any"))]
```

```
HUR_PTBP1_PTBP1HUR_overlaps <- HUR_PTBP1[as.data.frame(findOverlaps(PTBP1_GRanges,HUR_PTBP1_GRanges,type="any"))]
```

```
PTBP1_HURPTBP1_overlaps <- PTBP1[as.data.frame(findOverlaps(PTBP1_GRanges,HUR_PTBP1_GRanges,type="any"))]
```

```
HUR_HURPTBP1_overlaps <- HUR[as.data.frame(findOverlaps(HUR_GRanges,HUR_PTBP1_GRanges,type="any"))[,1],]
```

```
PTBP1_HURPTBP1_overlaps_adjust <- (PTBP1_HURPTBP1_overlaps$end - PTBP1_HURPTBP1_overlaps$start) - 35
```

```
HUR_PTBP1_PTBP1HUR_overlaps_adjust <- (HUR_PTBP1_PTBP1HUR_overlaps$end-HUR_PTBP1_PTBP1HUR_overlaps$start) - 35
```

```
HUR_HURPTBP1_overlaps_adjust <- (HUR_HURPTBP1_overlaps$end - HUR_HURPTBP1_overlaps$start) - 50
```

```
HUR_PTBP1_HURPTBP1_overlaps_adjust <- (HUR_PTBP1_HURPTBP1_overlaps$end-HUR_PTBP1_HURPTBP1_overlaps$start) - 35
```

```
All_overlaps_adjust_list <- list(HUR_HURPTBP1_overlaps_adjust,HUR_PTBP1_HURPTBP1_overlaps_adjust,NULL,PTBP1_HURPTBP1_overlaps_adjust)
names(All_overlaps_adjust_list) <- c("HUR","coRAP",NA,"PTBP1","coRAP")
```

```
#pdf("./Figure/Figure4/Figure4E_boxplot.pdf", width = 6, height = 7)
```

```
par(bty="n")
```

```
boxplot(All_overlaps_adjust_list, outline=F, ylab="Peak width normalized by lib size", las=2, ylim=c(-50,50))
```



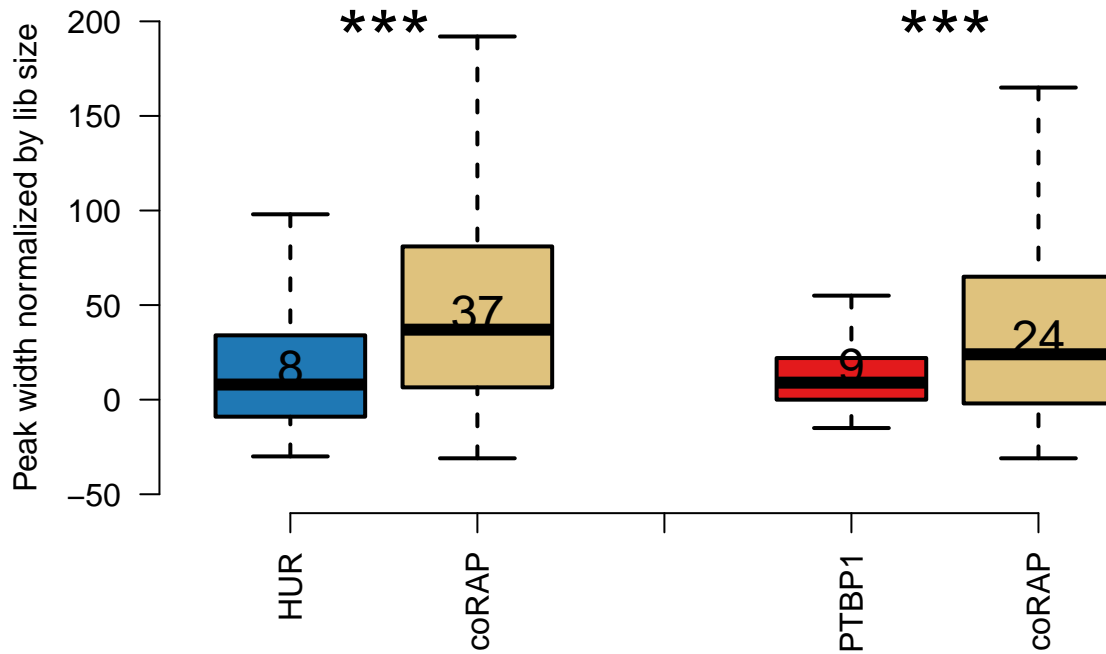
```

text(x=1,y=median(HUR_HURPTBP1_overlaps_adjust)+9, label=median(HUR_HURPTBP1_overlaps_adjust), cex=1.5)
text(x=2,y=median(HUR_PTBP1_HURPTBP1_overlaps_adjust)+9, label=median(HUR_PTBP1_HURPTBP1_overlaps_adjust), cex=1.5)
text(x=4,y=median(PTBP1_HURPTBP1_overlaps_adjust)+9, label=median(PTBP1_HURPTBP1_overlaps_adjust), cex=1.5)
text(x=5,y=median(HUR_PTBP1_PTBP1HUR_overlaps_adjust)+9, label=median(HUR_PTBP1_PTBP1HUR_overlaps_adjust), cex=1.5)

pw <- wilcox.test(PTBP1_HURPTBP1_overlaps_adjust, HUR_PTBP1_PTBP1HUR_overlaps_adjust, alternative = "less")
pwp <- pw$p.value
if ( pwp <= 0.001 ) {
  to_add <- "***"
} else if ( btp <= 0.01 ) {
  to_add <- "**"
} else if ( btp <= 0.05 ) {
  to_add <- "*"
} else {
  to_add <- "n.s."
}
text(1.5,200,labels = to_add, cex=2.5)

pw <- wilcox.test(HUR_HURPTBP1_overlaps_adjust, HUR_PTBP1_HURPTBP1_overlaps_adjust, alternative = "less")
pwp <- pw$p.value
if ( pwp <= 0.001 ) {
  to_add <- "***"
} else if ( btp <= 0.01 ) {
  to_add <- "**"
} else if ( btp <= 0.05 ) {
  to_add <- "*"
} else {
  to_add <- "n.s."
}
text(4.5,200,labels = to_add, cex=2.5)

```



```
#dev.off()
```

0.5.4 Figure 4F

```
HUR_PTBP1_uniq_4F <- unique(HUR_PTBP1[-as.data.frame(findOverlaps(HUR_GRanges, HUR_PTBP1_GRanges,type =

HUR_PTBP1_uniq_4F_select <- HUR_PTBP1_uniq_4F[,c(1,7,8,4:6)]
HUR_PTBP1_uniq_4F_select$Summit_start <- HUR_PTBP1_uniq_4F_select$Summit_start - 50
HUR_PTBP1_uniq_4F_select$Summit_end <- HUR_PTBP1_uniq_4F_select$Summit_end + 50
HUR_PTBP1_uniq_4F_select_adjust <- makeGRangesFromDataFrame(HUR_PTBP1_uniq_4F_select)

HUR_PTBP1_uniq_4F_overlap <- unique(HUR_PTBP1_uniq_4F[-as.data.frame(findOverlaps(PTBP1_GRanges, HUR_PTBP1_GRanges,type =

HUR_PTBP1_uniq_4F_overlap$positive_fa <- str_sub(HUR_PTBP1_uniq_4F_overlap$positive_fa,43,157)

bases <- c("A","C","T","G")
kmers_3 <- unite(as.data.frame(permutations(n=4,r=3,v=bases,repeats.allowed = T)),col = kmers,sep = "")
TTT_kmers_3 <- paste("TTT",kmers_3,sep="")
TTT_kmers_3 <- paste(TTT_kmers_3,"TTT",sep="")
kmers_2 <- unite(as.data.frame(permutations(n=4,r=2,v=bases,repeats.allowed = T)),col = kmers,sep = "")
TTT_kmers_2 <- paste("TTT",kmers_2,sep="")
TTT_kmers_2 <- paste(TTT_kmers_2,"TTT",sep="")
TTT_kmers_1 <- c("TTTTTTTT","TTTATTT","TTTGTTT","TTTCCTTT")
```

```

hur_kmers <- c(TTT_kmers_1, TTT_kmers_2, TTT_kmers_3)

ptbp1_kmers <- c("TCTCT", "TTTCT", "TTTCT", "CTTCT", "TCTCT", "CCTCT", "GTTCT", "CTTCT", "GCTCT", "CCTCT", "GTTCT")

for (i in hur_kmers){
  HUR_PTBP1_uniq_4F_overlap$positive_fa <- gsub(i, "U", HUR_PTBP1_uniq_4F_overlap$positive_fa)
}

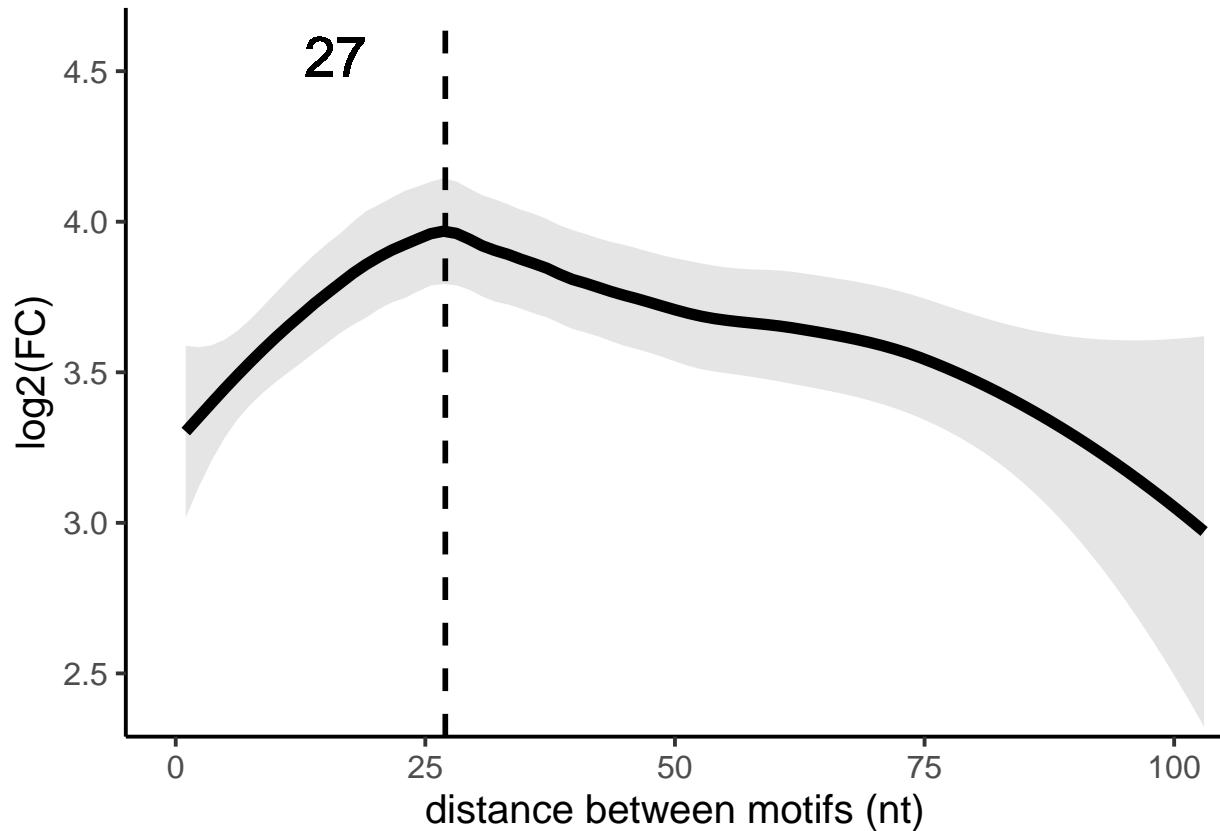
for (i in ptbp1_kmers){
  HUR_PTBP1_uniq_4F_overlap$positive_fa <- gsub(i, "Y", HUR_PTBP1_uniq_4F_overlap$positive_fa)
}

HUR_PTBP1_uniq_4F_overlap$Ys <- str_count(HUR_PTBP1_uniq_4F_overlap$positive_fa, "Y")
HUR_PTBP1_uniq_4F_overlap$Us <- str_count(HUR_PTBP1_uniq_4F_overlap$positive_fa, "U")
HUR_PTBP1_uniq_4F_overlap$TTTs <- str_count(HUR_PTBP1_uniq_4F_overlap$positive_fa, "TTT")

HUR_PTBP1_uniq_4F_overlap_YsUs <- HUR_PTBP1_uniq_4F_overlap[HUR_PTBP1_uniq_4F_overlap$Ys == 1 & HUR_PTBP1_uniq_4F_overlap$Us == 1, ]
HUR_PTBP1_uniq_4F_overlap_YsUs$Y_loc <- unlist(str_locate_all(HUR_PTBP1_uniq_4F_overlap_YsUs$positive_fa, "Y"))
HUR_PTBP1_uniq_4F_overlap_YsUs$U_loc <- unlist(str_locate_all(HUR_PTBP1_uniq_4F_overlap_YsUs$positive_fa, "U"))
HUR_PTBP1_uniq_4F_overlap_YsUs$distance <- abs(HUR_PTBP1_uniq_4F_overlap_YsUs$Y_loc - HUR_PTBP1_uniq_4F_overlap_YsUs$U_loc)

HUR_PTBP1_uniq_4F_overlap_YsUs_plot <- ggplot2::ggplot(data=HUR_PTBP1_uniq_4F_overlap_YsUs, aes(x=distance, y=log2(FC))) +
  stat_smooth(method="loess", color="black", fill="grey", span=0.6, level=0.95, lwd=2) +
  theme_classic2(base_size = 15) +
  coord_cartesian(xlim=c(0,100), ylim=c(2.4,4.6)) +
  geom_vline(xintercept = c(27), color="black", lty=2, lwd=1) +
  geom_text(aes(x=16, y=4.55), label = 27, size=7.5, col="black") +
  xlab("distance between motifs (nt)") +
  ylab("log2(FC)")
pdf("./Figure/Figure4/Figure4F.pdf", width = 6, height = 6)
HUR_PTBP1_uniq_4F_overlap_YsUs_plot
## Warning in geom_text(aes(x = 16, y = 4.55), label = 27, size = 7.5, col = "black"): All aesthetics have been used
## i Please consider using `annotate()` or provide this layer with data containing
## a single row.
## `geom_smooth()` using formula = 'y ~ x'

```



```
#dev.off()
```

0.5.5 Figure S4A

```
HUR_PTBP1_uniq_S4A <- unique(HUR_PTBP1[-as.data.frame(findOverlaps(HUR_GRanges, HUR_PTBP1_GRanges,type = "any"))])

HUR_PTBP1_uniq_S4A_select <- HUR_PTBP1_uniq_S4A[,c(1,7,8,4:6)]
HUR_PTBP1_uniq_S4A_select$Summit_start <- HUR_PTBP1_uniq_S4A_select$Summit_start - 50
HUR_PTBP1_uniq_S4A_select$Summit_end <- HUR_PTBP1_uniq_S4A_select$Summit_end + 50
HUR_PTBP1_uniq_S4A_select_GRanges <- makeGRangesFromDataFrame(HUR_PTBP1_uniq_S4A_select)

HUR_PTBP1_uniq_S4A_overlap <- unique(HUR_PTBP1_uniq_S4A[-as.data.frame(findOverlaps(PTBP1_GRanges, HUR_PTBP1_uniq_S4A_GRanges,type = "any"))])

HUR_PTBP1_uniq_S4A_overlap$negative_fa <- str_sub(HUR_PTBP1_uniq_S4A_overlap$negative_fa,43,157)

bases <- c("A","C","T","G")
kmers_3 <- unite(as.data.frame(permutations(n=4,r=3,v=bases,repeats.allowed = T)),col = kmers,sep = "")
TTT_kmers_3 <- paste("TTT",kmers_3,sep="")
TTT_kmers_3 <- paste(TTT_kmers_3,"TTT",sep="")
kmers_2 <- unite(as.data.frame(permutations(n=4,r=2,v=bases,repeats.allowed = T)),col = kmers,sep = "")
TTT_kmers_2 <- paste("TTT",kmers_2,sep="")
TTT_kmers_2 <- paste(TTT_kmers_2,"TTT",sep="")
TTT_kmers_1 <- c("TTTTTTT","TTTATTT","TTTGTTT","TTTCTTT")
```

```

hur_kmers <- c(TTT_kmers_1,TTT_kmers_2,TTT_kmers_3)
ptbp1_kmers <- c("TCTCT","TTTCT","TTTCT","CTTCT","TCTCT","CCTCT","GTTCT","CTTCT","GCTCT","CCTCT","GTTCT")

for (i in hur_kmers){
  HUR_PTBP1_uniq_S4A_overlap$negative_fa <- gsub(i,"U",HUR_PTBP1_uniq_S4A_overlap$negative_fa)
}

for (i in ptbp1_kmers){
  HUR_PTBP1_uniq_S4A_overlap$negative_fa <- gsub(i,"Y",HUR_PTBP1_uniq_S4A_overlap$negative_fa)
}

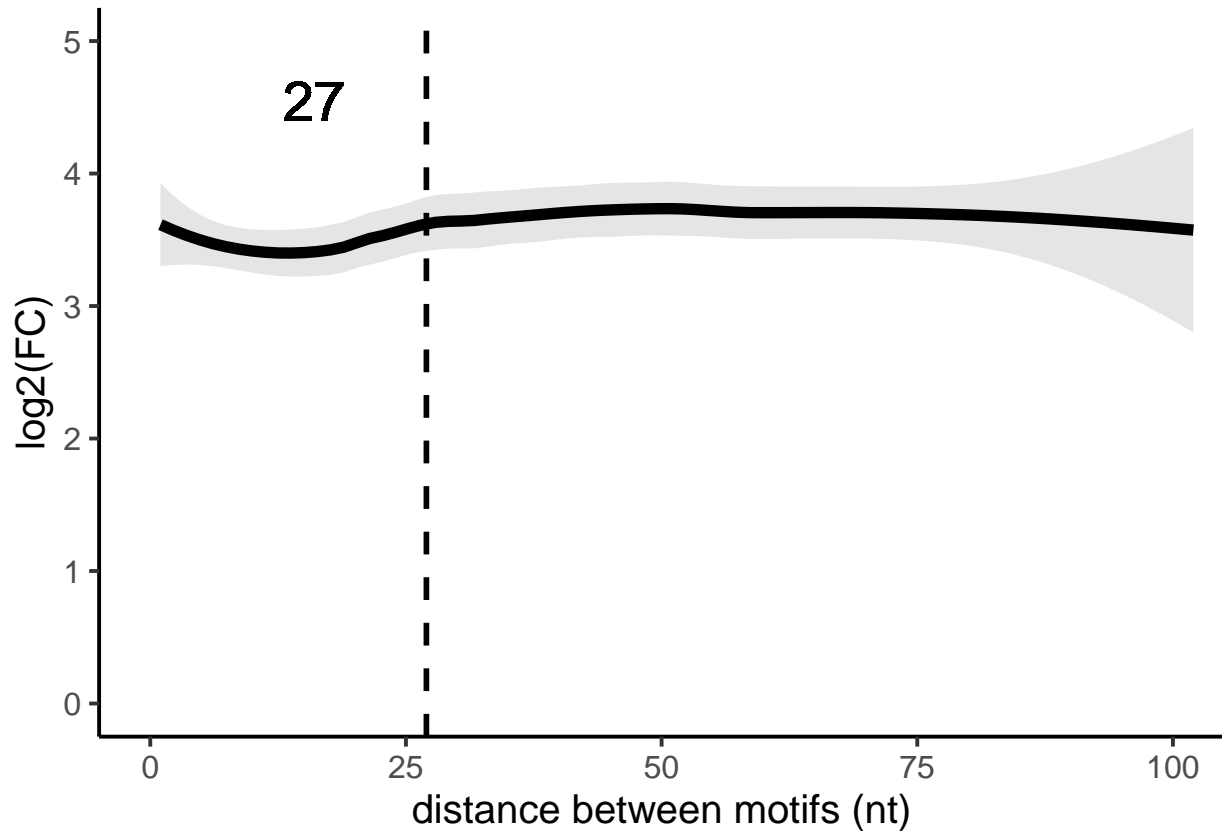
HUR_PTBP1_uniq_S4A_overlap$Ys <- str_count(HUR_PTBP1_uniq_S4A_overlap$negative_fa,"Y")
HUR_PTBP1_uniq_S4A_overlap$Us <- str_count(HUR_PTBP1_uniq_S4A_overlap$negative_fa,"U")
HUR_PTBP1_uniq_S4A_overlap$TTTs <- str_count(HUR_PTBP1_uniq_S4A_overlap$negative_fa,"TTT")

HUR_PTBP1_uniq_S4A_overlap_YsUs <- HUR_PTBP1_uniq_S4A_overlap[HUR_PTBP1_uniq_S4A_overlap$Ys == 1 & HUR_PTBP1_uniq_S4A_overlap$Us == 1,]
HUR_PTBP1_uniq_S4A_overlap_YsUs$Y_loc <- unlist(str_locate_all(HUR_PTBP1_uniq_S4A_overlap_YsUs$negative_fa,"Y"))
HUR_PTBP1_uniq_S4A_overlap_YsUs$U_loc <- unlist(str_locate_all(HUR_PTBP1_uniq_S4A_overlap_YsUs$negative_fa,"U"))
HUR_PTBP1_uniq_S4A_overlap_YsUs$distance <- abs(HUR_PTBP1_uniq_S4A_overlap_YsUs$Y_loc - HUR_PTBP1_uniq_S4A_overlap_YsUs$U_loc)

HUR_PTBP1_uniq_S4A_overlap_YsUs_plot <- ggplot2::ggplot(data=HUR_PTBP1_uniq_S4A_overlap_YsUs, aes(x=distance, y=log2(FC))) +
  stat_smooth(method="loess", color="black", fill="grey", span=0.6, level=0.95, lwd=2) +
  theme_classic2(base_size = 15) +
  coord_cartesian(xlim=c(0,100), ylim=c(0, 5)) +
  geom_vline(xintercept = c(27), color="black", lty=2, lwd=1) +
  geom_text(aes(x=16,y=4.55), label = 27, size=7.5, col="black") +
  xlab("distance between motifs (nt)") +
  ylab("log2(FC)")

#pdf("./Figure/Figure4/FigureS4A.pdf", width = 6, height = 6)
HUR_PTBP1_uniq_S4A_overlap_YsUs_plot
## Warning in geom_text(aes(x = 16, y = 4.55), label = 27, size = 7.5, col = "black"): All aesthetics have been overridden
## i Please consider using `annotate()` or provide this layer with data containing
##   a single row.
## `geom_smooth()` using formula = 'y ~ x'

```



```
#dev.off()
```

0.5.6 Figure S4B

```
HUR_uniq_S4B <- unique(HUR[-as.data.frame(findOverlaps(HUR_GRanges,HUR_PTBP1_GRanges,type = "any"))[,1]]

HUR_uniq_S4B_select <- HUR_uniq_S4B[,c(1,7,8,4:6)]
HUR_uniq_S4B_select$Summit_start <- HUR_uniq_S4B_select$Summit_start - 50
HUR_uniq_S4B_select$Summit_end <- HUR_uniq_S4B_select$Summit_end + 50
HUR_uniq_S4B_select_GRanges <- makeGRangesFromDataFrame(HUR_uniq_S4B_select)

HUR_uniq_S4B_overlaps <- unique(HUR_uniq_S4B[-as.data.frame(findOverlaps(PTBP1_GRanges, HUR_uniq_S4B_select_GRanges))[,1]]

HUR_uniq_S4B_overlaps$positive_fa <- str_sub(HUR_uniq_S4B_overlaps$positive_fa,43,157)

bases <- c("A","C","T","G")
kmers_3 <- unite(as.data.frame(permutations(n=4,r=3,v=bases,repeats.allowed = T)),col = kmers,sep = "")
TTT_kmers_3 <- paste("TTT",kmers_3,sep="")
TTT_kmers_3 <- paste(TTT_kmers_3,"TTT",sep="")
kmers_2 <- unite(as.data.frame(permutations(n=4,r=2,v=bases,repeats.allowed = T)),col = kmers,sep = "")
TTT_kmers_2 <- paste("TTT",kmers_2,sep="")
TTT_kmers_2 <- paste(TTT_kmers_2,"TTT",sep="")
```

```

TTT_kmers_1 <- c("TTTTTTT", "TTTATTT", "TTTGTTT", "TTTCTTT")
hur_kmers <- c(TTT_kmers_1, TTT_kmers_2, TTT_kmers_3)
ptbpl_kmers <- c("TCTCT", "TTTCT", "TTTCT", "CTTCT", "TCTCT", "CCTCT", "GTTCT", "CTTCT", "GCTCT", "CCTCT", "GTTCT")

for (i in hur_kmers){
  HUR_uniq_S4B_overlaps$positive_fa <- gsub(i, "U", HUR_uniq_S4B_overlaps$positive_fa)
}

for (i in ptbpl_kmers){
  HUR_uniq_S4B_overlaps$positive_fa <- gsub(i, "Y", HUR_uniq_S4B_overlaps$positive_fa)
}

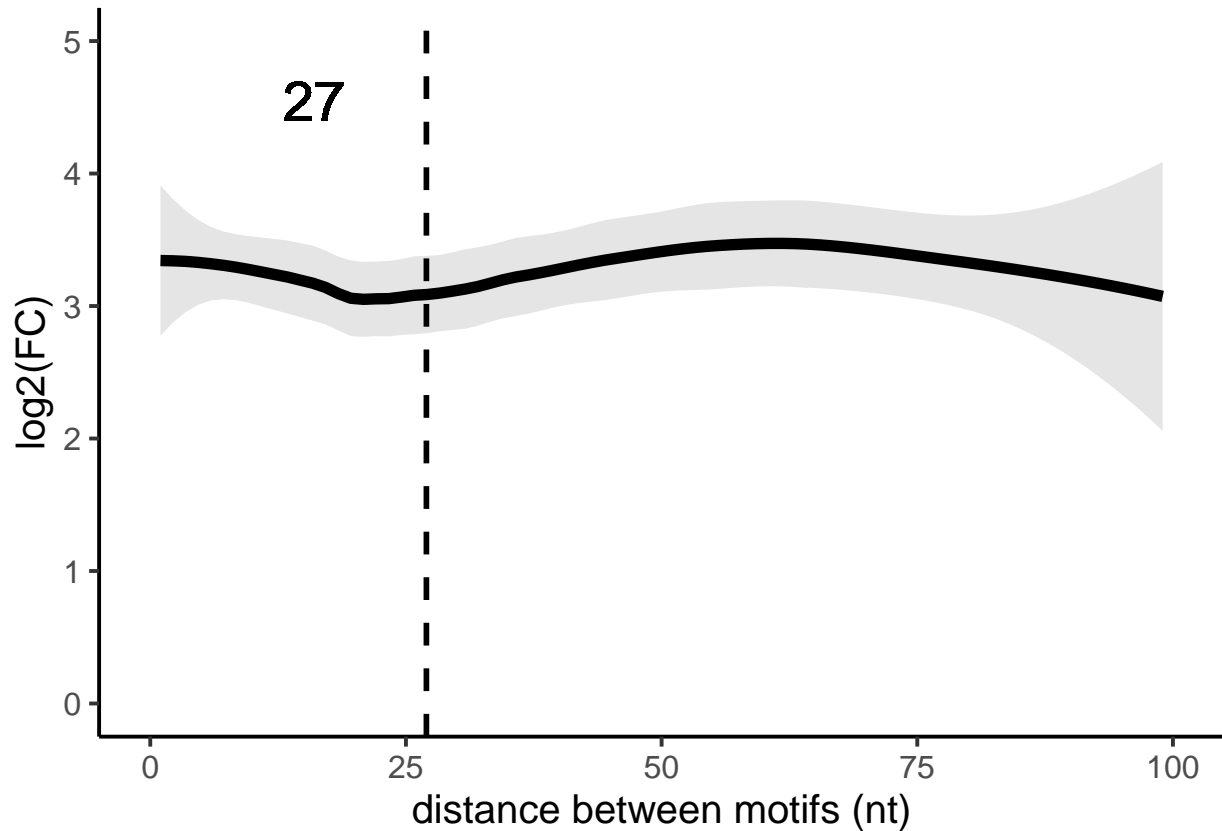
HUR_uniq_S4B_overlaps$Ys <- str_count(HUR_uniq_S4B_overlaps$positive_fa, "Y")
HUR_uniq_S4B_overlaps$Us <- str_count(HUR_uniq_S4B_overlaps$positive_fa, "U")
HUR_uniq_S4B_overlaps$TTTs <- str_count(HUR_uniq_S4B_overlaps$positive_fa, "TTT")

HUR_uniq_S4B_overlaps_YsUs <- HUR_uniq_S4B_overlaps[HUR_uniq_S4B_overlaps$Ys == 1 & HUR_uniq_S4B_overlaps$Us == 1, ]
HUR_uniq_S4B_overlaps_YsUs$Y_loc <- unlist(str_locate_all(HUR_uniq_S4B_overlaps_YsUs$positive_fa, "Y"))[, 2]
HUR_uniq_S4B_overlaps_YsUs$U_loc <- unlist(str_locate_all(HUR_uniq_S4B_overlaps_YsUs$positive_fa, "U"))[, 2]
HUR_uniq_S4B_overlaps_YsUs$distance <- abs(HUR_uniq_S4B_overlaps_YsUs$Y_loc - HUR_uniq_S4B_overlaps_YsUs$U_loc)

HUR_uniq_S4B_overlaps_YsUs_plot <- ggplot2::ggplot(data=HUR_uniq_S4B_overlaps_YsUs, aes(x=distance, y=log2(FC))) +
  stat_smooth(method="loess", color="black", fill="grey", span=0.6, level=0.95, lwd=2) +
  theme_classic2(base_size = 15) +
  coord_cartesian(xlim=c(0,100), ylim=c(0,5)) +
  geom_vline(xintercept = c(27), color="black", lty=2, lwd=1) +
  geom_text(aes(x=16, y=4.55), label = 27, size=7.5, col="black") +
  xlab("distance between motifs (nt)") +
  ylab("log2(FC)")

#pdf("./Figure/Figure4/FigureS4B.pdf", width = 6, height = 6)
HUR_uniq_S4B_overlaps_YsUs_plot
## Warning in geom_text(aes(x = 16, y = 4.55), label = 27, size = 7.5, col = "black"): All aesthetics have been used
## i Please consider using `annotate()` or provide this layer with data containing
## a single row.
## `geom_smooth()` using formula = 'y ~ x'

```



```
#dev.off()
```

0.5.7 Figure S4C

```
PTBP1_uniq_S4C <- unique(PTBP1[-as.data.frame(findOverlaps(PTBP1_GRanges,HUR_PTBP1_GRanges,type = "any"))])

PTBP1_uniq_S4C_select <- PTBP1_uniq_S4C[,c(1,7,8,4:6)]
PTBP1_uniq_S4C_select$Summit_start <- PTBP1_uniq_S4C_select$Summit_start - 50
PTBP1_uniq_S4C_select$Summit_end <- PTBP1_uniq_S4C_select$Summit_end + 50
PTBP1_uniq_S4C_select_GRanges <- makeGRangesFromDataFrame(PTBP1_uniq_S4C_select)

PTBP1_uniq_S4C_select_GRanges_overlaps <- unique(PTBP1_uniq_S4C[-as.data.frame(findOverlaps(HUR_GRanges,PTBP1_uniq_S4C_select_GRanges,type = "any"))])

PTBP1_uniq_S4C_select_GRanges_overlaps$positive_fa <- str_sub(PTBP1_uniq_S4C_select_GRanges_overlaps$positive_fa,1,10)

bases <- c("A","C","T","G")
kmers_3 <- unite(as.data.frame(permutations(n=4,r=3,v=bases,repeats.allowed = T)),col = kmers,sep = "")
TTT_kmers_3 <- paste("TTT",kmers_3,sep="")
TTT_kmers_3 <- paste(TTT_kmers_3,"TTT",sep="")
kmers_2 <- unite(as.data.frame(permutations(n=4,r=2,v=bases,repeats.allowed = T)),col = kmers,sep = "")
TTT_kmers_2 <- paste("TTT",kmers_2,sep="")
TTT_kmers_2 <- paste(TTT_kmers_2,"TTT",sep="")
```



```

TTT_kmers_1 <- c("TTTTTTT", "TTTATTT", "TTTGTTT", "TTTCTTT")
hur_kmers <- c(TTT_kmers_1, TTT_kmers_2, TTT_kmers_3)
ptbp1_kmers <- c("TCTCT", "TTTCT", "TTTCT", "CTTCT", "TCTCT", "CCTCT", "GTTCT", "CTTCT", "GCTCT", "CCTCT", "GTTCT")

for (i in hur_kmers){
  PTBP1_uniq_S4C_select_GRanges_overlaps$positive_fa <- gsub(i, "U", PTBP1_uniq_S4C_select_GRanges_overlaps$positive_fa)
}

for (i in ptbp1_kmers){
  PTBP1_uniq_S4C_select_GRanges_overlaps$positive_fa <- gsub(i, "Y", PTBP1_uniq_S4C_select_GRanges_overlaps$positive_fa)
}

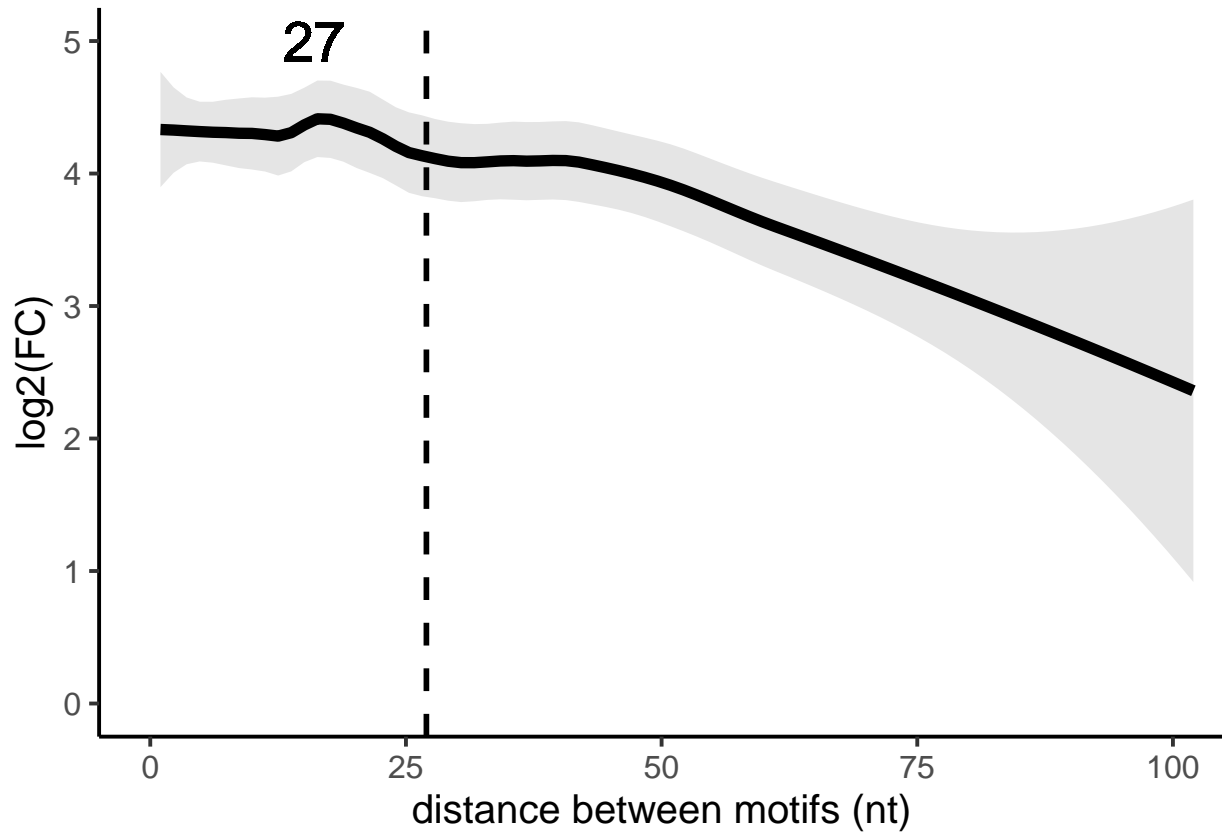
PTBP1_uniq_S4C_select_GRanges_overlaps$Ys <- str_count(PTBP1_uniq_S4C_select_GRanges_overlaps$positive_fa, "Y")
PTBP1_uniq_S4C_select_GRanges_overlaps$Us <- str_count(PTBP1_uniq_S4C_select_GRanges_overlaps$positive_fa, "U")
PTBP1_uniq_S4C_select_GRanges_overlaps$TTTs <- str_count(PTBP1_uniq_S4C_select_GRanges_overlaps$positive_fa, "T")

PTBP1_uniq_S4C_select_GRanges_overlaps_YsUs <- PTBP1_uniq_S4C_select_GRanges_overlaps[PTBP1_uniq_S4C_select_GRanges_overlaps$Ys > 0 && PTBP1_uniq_S4C_select_GRanges_overlaps$Us > 0, ]
PTBP1_uniq_S4C_select_GRanges_overlaps_YsUs$Y_loc <- unlist(str_locate_all(PTBP1_uniq_S4C_select_GRanges_overlaps_YsUs$positive_fa, "Y"))
PTBP1_uniq_S4C_select_GRanges_overlaps_YsUs$U_loc <- unlist(str_locate_all(PTBP1_uniq_S4C_select_GRanges_overlaps_YsUs$positive_fa, "U"))
PTBP1_uniq_S4C_select_GRanges_overlaps_YsUs$distance <- abs(PTBP1_uniq_S4C_select_GRanges_overlaps_YsUs$Y_loc - PTBP1_uniq_S4C_select_GRanges_overlaps_YsUs$U_loc)

PTBP1_uniq_S4C_select_GRanges_overlaps_YsUs_plot <- ggplot2::ggplot(data=PTBP1_uniq_S4C_select_GRanges_overlaps_YsUs, aes(x=distance, y=log2(FC))) +
  stat_smooth(method="loess", color="black", fill="grey", span=0.6, level=0.95, lwd=2) +
  theme_classic2(base_size = 15) +
  coord_cartesian(xlim=c(0,100), ylim=c(0,5)) +
  geom_vline(xintercept = c(27), color="black", lty=2, lwd=1) +
  geom_text(aes(x=16, y=5), label = 27, size=7.5, col="black") +
  xlab("distance between motifs (nt)") +
  ylab("log2(FC)")

#pdf("./Figure/Figure4/FigureS4C.pdf", width = 6, height = 6)
PTBP1_uniq_S4C_select_GRanges_overlaps_YsUs_plot
## Warning in geom_text(aes(x = 16, y = 5), label = 27, size = 7.5, col = "black"): All aesthetics have
## i Please consider using `annotate()` or provide this layer with data containing
## a single row.
## `geom_smooth()` using formula = 'y ~ x'

```



```
#dev.off()
```

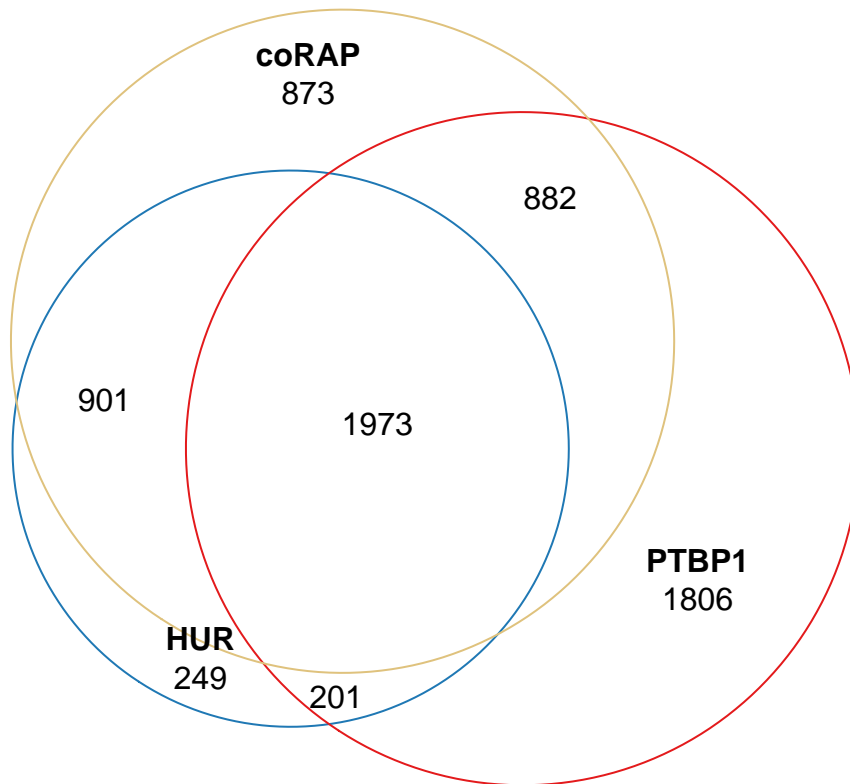
0.5.8 Figure S4D

```
PTBP1_gene <- unique(PTBP1[, "gene_name"])
HUR_gene <- unique(HUR[, "gene_name"])
coRAP_gene <- unique(HUR_PTBP1[, "gene_name"])
HUR_PTBP1_coRAP_gene <- list(HUR_gene, PTBP1_gene, coRAP_gene)

names(HUR_PTBP1_coRAP_gene) <- c("HUR", "PTBP1", "coRAP")
HUR_PTBP1_coRAP_gene_overlap <- euler(HUR_PTBP1_coRAP_gene, shape="circle")

#pdf("../Figure/Figure4/FigureS4D.pdf", width = 6, height = 6)
plot(HUR_PTBP1_coRAP_gene_overlap, fills=c("white", "white", "white"), quantities=TRUE, edges=T, col=c("#
```

gene overlaps



```
#dev.off()
```

```
###Figure S4E left
```

```
# load data
HUR_coRAP_gene <- as.data.frame(intersect(HUR_gene, coRAP_gene))
colnames(HUR_coRAP_gene) <- "gene_name"

HUR_coRAP_data <- merge(HUR, HUR_coRAP_gene, by="gene_name")
HUR_coRAP_data <- HUR_coRAP_data[,c("Rep1", "Rep2", "gene_name")]
HUR_coRAP_data_melt <- melt(HUR_coRAP_data) %>% group_by(gene_name, variable) %>% summarise(Gene_Counts = sum(Rep1, Rep2))
## Using gene_name as id variables
## `summarise()` has grouped output by 'gene_name'. You can override using the
## `.groups` argument.
HUR_coRAP_data_melt <- as.data.frame(HUR_coRAP_data_melt)
HUR_coRAP_data <- reshape2::dcast(HUR_coRAP_data_melt, gene_name~variable)
## Using Gene_Counts as value column: use value.var to override.
colnames(HUR_coRAP_data) <- c("gene_name", "HUR_1", "HUR_2")

HURPTBP1_coRAP_data <- merge(HUR_PTBP1, HUR_coRAP_gene, by="gene_name")
HURPTBP1_coRAP_data <- HURPTBP1_coRAP_data[,c("Rep1", "Rep2", "gene_name")]
HURPTBP1_coRAP_data_melt <- melt(HURPTBP1_coRAP_data) %>% group_by(gene_name, variable) %>% summarise(Gene_Counts = sum(Rep1, Rep2))
## Using gene_name as id variables
## `summarise()` has grouped output by 'gene_name'. You can override using the `.groups` argument.
HURPTBP1_coRAP_data_melt <- as.data.frame(HURPTBP1_coRAP_data_melt)
```

```

HURPTBP1_coRAP_data <- reshape2::dcast(HURPTBP1_coRAP_data_melt, gene_name~variable)
## Using Gene_Counts as value column: use value.var to override.
colnames(HURPTBP1_coRAP_data) <- c("gene_name", "coRAP_1", "coRAP_2")

HUR_coRAP_commongenes <- merge(HUR_coRAP_data, HURPTBP1_coRAP_data, by="gene_name")

# DEG analysis
x <- HUR_coRAP_commongenes[,2:5]
group <- c(1,1,2,2)
y <- DGEList(counts=x, group = group)
design <- model.matrix(~group)
y <- estimateDisp(y, design)

fit <- glmQLFit(y, design)
qlf <- glmQLFTest(fit)

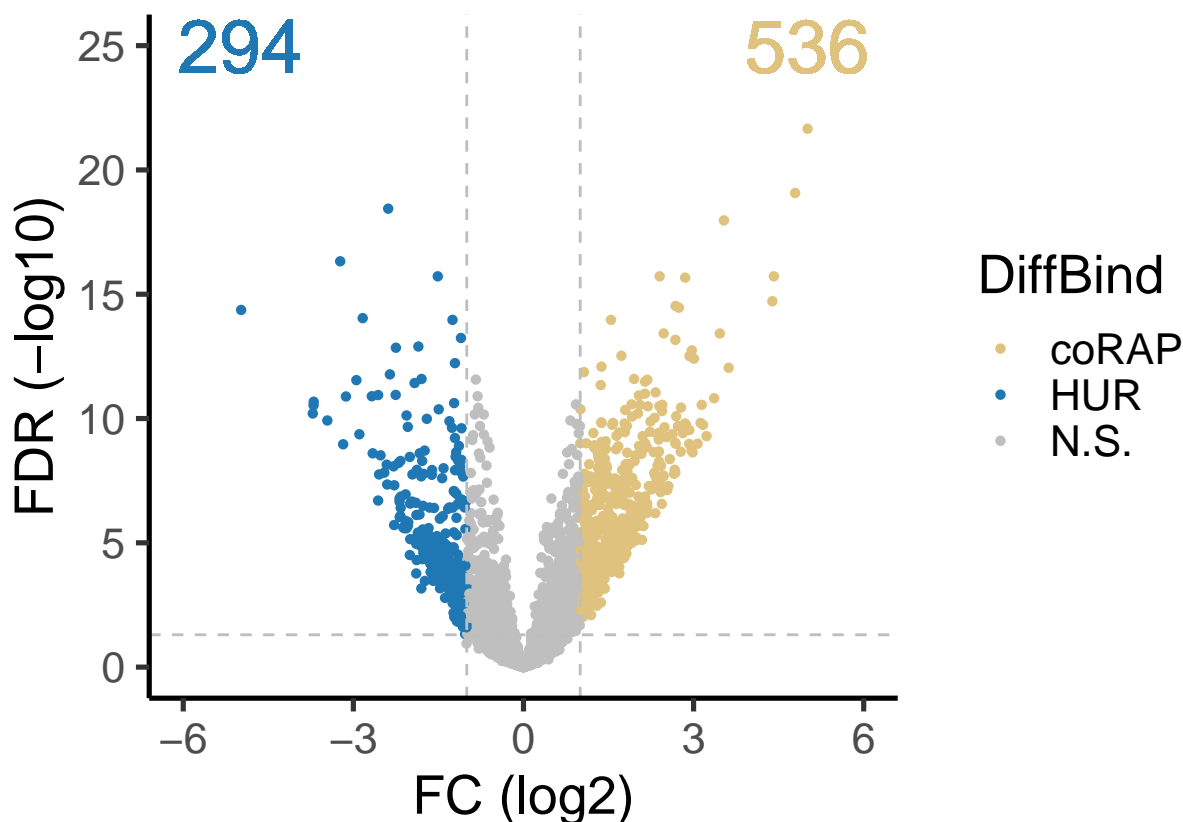
DEG_data_table_S4EL <- qlf$table
DEG_data_table_S4EL$p.adjust <- -log10(p.adjust(DEG_data_table_S4EL$PValue, method = "BH"))
DEG_data_table_S4EL$PValue <- -log10(DEG_data_table_S4EL$PValue)

DEG_data_table_S4EL$HUR_won <- DEG_data_table_S4EL$logFC <= -1 & DEG_data_table_S4EL$p.adjust >= 1.30103
DEG_data_table_S4EL$coRAP_won <- DEG_data_table_S4EL$logFC >= 1 & DEG_data_table_S4EL$p.adjust >= 1.30103
DEG_data_table_S4EL$DiffBind <- paste(DEG_data_table_S4EL$HUR_won, DEG_data_table_S4EL$coRAP_won, sep = ".")
DEG_data_table_S4EL$DiffBind <- gsub("FALSE_FALSE", "N.S.", DEG_data_table_S4EL$DiffBind)
DEG_data_table_S4EL$DiffBind <- gsub("FALSE_TRUE", "coRAP", DEG_data_table_S4EL$DiffBind)
DEG_data_table_S4EL$DiffBind <- gsub("TRUE_FALSE", "HUR", DEG_data_table_S4EL$DiffBind)

DEG_data_table_S4EL_plot <- ggplot(data=DEG_data_table_S4EL) +
  geom_point(aes(x=logFC, y=p.adjust, color=DiffBind), pch=16) +
  geom_hline(yintercept = -log10(0.05), linetype = "dashed", color = "grey") +
  geom_vline(xintercept = c(-1,1), linetype = "dashed", color = "grey") +
  scale_color_manual(values = c("#dfc27d", "#1f78b4", "grey75")) +
  theme_classic(base_size = 20) +
  ylab("FDR (-log10)") +
  xlab("FC (log2)") +
  ylim(0,25) +
  xlim(-6,6)
DEG_data_table_S4EL_plot <- DEG_data_table_S4EL_plot + geom_text(aes(x=5, y=25), label = table(DEG_data_table_S4EL$DiffBind)[1,2]) +
  geom_text(aes(x=-5, y=25), label = table(DEG_data_table_S4EL$DiffBind)[2,1], size=10, color="#1f78b4")

#pdf("./Figure/Figure4/FigureS4E_left.pdf", width = 6, height = 6)
DEG_data_table_S4EL_plot
## Warning in geom_text(aes(x = 5, y = 25), label = table(DEG_data_table_S4EL$DiffBind)[1,2], : All
## i Please consider using `annotate()` or provide this layer with data containing
##   a single row.
## Warning in geom_text(aes(x = -5, y = 25), label = table(DEG_data_table_S4EL$DiffBind)[2,1], : All
## i Please consider using `annotate()` or provide this layer with data containing
##   a single row.

```



```
#dev.off()
```

0.5.9 Figure S4E right

```
PTBP1_coRAP_gene <- as.data.frame(intersect(PTBP1_gene, coRAP_gene))
colnames(PTBP1_coRAP_gene) <- "gene_name"

PTBP1_coRAP_data <- merge(PTBP1, PTBP1_coRAP_gene, by="gene_name")
PTBP1_coRAP_data <- PTBP1_coRAP_data[, c("Rep1", "Rep2", "gene_name")]
PTBP1_coRAP_data_melt <- melt(PTBP1_coRAP_data) %>% group_by(gene_name, variable) %>% summarise(Gene_Count = sum(value))
## Using gene_name as id variables
## `summarise()` has grouped output by 'gene_name'. You can override using the `groups` argument.

PTBP1_coRAP_data_melt <- as.data.frame(PTBP1_coRAP_data_melt)
PTBP1_coRAP_data <- reshape2::dcast(PTBP1_coRAP_data_melt, gene_name ~ variable)
## Using Gene_Counts as value column: use value.var to override.
colnames(PTBP1_coRAP_data) <- c("gene_name", "PTBP1_1", "PTBP1_2")

HURPTBP1_coRAP_data <- merge(HUR_PTBP1, PTBP1_coRAP_gene, by="gene_name")
HURPTBP1_coRAP_data <- HURPTBP1_coRAP_data[, c("Rep1", "Rep2", "gene_name")]
HURPTBP1_coRAP_data_melt <- melt(HURPTBP1_coRAP_data) %>% group_by(gene_name, variable) %>% summarise(Gene_Count = sum(value))
## Using gene_name as id variables
## `summarise()` has grouped output by 'gene_name'. You can override using the `groups` argument.
HURPTBP1_coRAP_data_melt <- as.data.frame(HURPTBP1_coRAP_data_melt)
```

```

HURPTBP1_coRAP_data <- reshape2::dcast(HURPTBP1_coRAP_data_melt, gene_name~variable)
## Using Gene_Counts as value column: use value.var to override.
colnames(HURPTBP1_coRAP_data) <- c("gene_name", "coRAP_1", "coRAP_2")

HUR_coRAP_commongenes <- merge(PTBP1_coRAP_data, HURPTBP1_coRAP_data, by="gene_name")

# DEG analysis
x <- HUR_coRAP_commongenes[,2:5]
group <- c(1,1,2,2)
y <- DGEList(counts=x, group = group)
design <- model.matrix(~group)
y <- estimateDisp(y, design)

fit <- glmQLFit(y, design)
qlf <- glmQLFTest(fit)

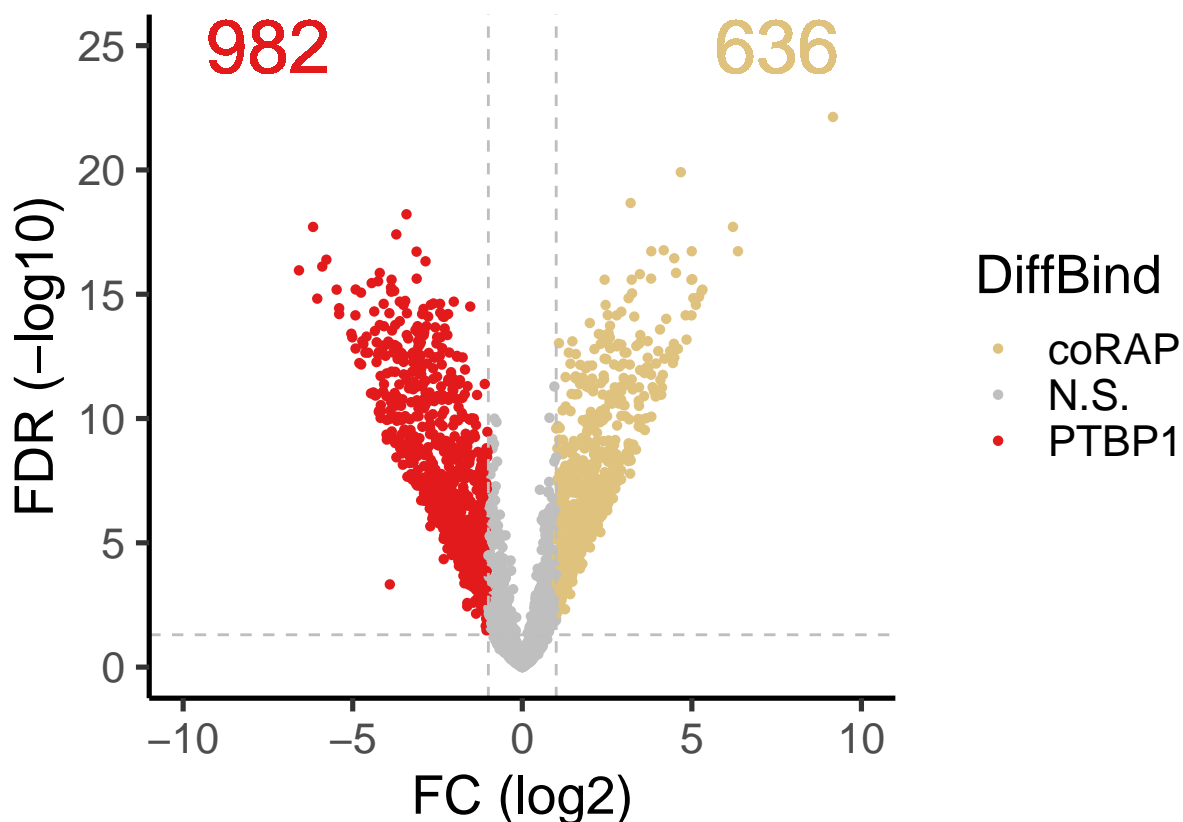
DEG_data_table_S4ER <- qlf$table
DEG_data_table_S4ER$p.adjust <- -log10(p.adjust(DEG_data_table_S4ER$PValue, method = "BH"))
DEG_data_table_S4ER$PValue <- -log10(DEG_data_table_S4ER$PValue)

DEG_data_table_S4ER$HUR_won <- DEG_data_table_S4ER$logFC <= -1 & DEG_data_table_S4ER$p.adjust >= 1.30103
DEG_data_table_S4ER$coRAP_won <- DEG_data_table_S4ER$logFC >= 1 & DEG_data_table_S4ER$p.adjust >= 1.30103
DEG_data_table_S4ER$DiffBind <- paste(DEG_data_table_S4ER$HUR_won, DEG_data_table_S4ER$coRAP_won, sep = ".")
DEG_data_table_S4ER$DiffBind <- gsub("FALSE_FALSE", "N.S.", DEG_data_table_S4ER$DiffBind)
DEG_data_table_S4ER$DiffBind <- gsub("FALSE_TRUE", "coRAP", DEG_data_table_S4ER$DiffBind)
DEG_data_table_S4ER$DiffBind <- gsub("TRUE_FALSE", "PTBP1", DEG_data_table_S4ER$DiffBind)

DEG_data_table_S4ER_plot <- ggplot(data=DEG_data_table_S4ER) +
  geom_point(aes(x=logFC, y=p.adjust, color=DiffBind), pch=16) +
  geom_hline(yintercept = -log10(0.05), linetype = "dashed", color = "grey") +
  geom_vline(xintercept = c(-1,1), linetype = "dashed", color = "grey") +
  scale_color_manual(values = c("#dfc27d", "grey75", "#e31a1c")) +
  theme_classic(base_size = 20) +
  ylab("FDR (-log10)") +
  xlab("FC (log2)") +
  ylim(0,25) +
  xlim(-10,10)
DEG_data_table_S4ER_plot <- DEG_data_table_S4ER_plot + geom_text(aes(x=7.5, y=25), label = table(DEG_data_table_S4ER$DiffBind)[1], size=10, color="#e31a1c")
  geom_text(aes(x=-7.5, y=25), label = table(DEG_data_table_S4ER$DiffBind)[2], size=10, color="#e31a1c")

#pdf("../Figure/Figure4/FigureS4E_right.pdf", width = 6, height = 6)
DEG_data_table_S4ER_plot
## Warning in geom_text(aes(x = 7.5, y = 25), label = table(DEG_data_table_S4ER$DiffBind)[1], : A single row.
## i Please consider using `annotate()` or provide this layer with data containing
## a single row.
## Warning in geom_text(aes(x = -7.5, y = 25), label = table(DEG_data_table_S4ER$DiffBind)[2], : A single row.
## i Please consider using `annotate()` or provide this layer with data containing
## a single row.

```



```
#dev.off()
```

0.5.10 Figure S4F

```
PTBP1_gene <- unique(PTBP1[, "gene_name"])
HUR_gene <- unique(HUR[, "gene_name"])
coRAP_gene <- unique(HUR_PTBP1[, "gene_name"])

coRAP_only <- setdiff(coRAP_gene, HUR_gene)
coRAP_only <- setdiff(coRAP_only, PTBP1_gene)

HUR_DB <- DEG_data_table_S4EL
coRAP_HU_diffb <- HUR_coRAP_commongenes[HUR_DB$DiffBind == "coRAP", "gene_name"]

PTBP1_DB <- DEG_data_table_S4ER
coRAP_PT_diffb <- HUR_coRAP_commongenes[PTBP1_DB$DiffBind == "coRAP", "gene_name"]

gene_uniq <- unique(c(coRAP_only, coRAP_HU_diffb, coRAP_PT_diffb))

entrez_IDS <- na.omit(as.data.frame(unlist(mapIds(org.Hs.eg.db, gene_uniq, 'ENTREZID', 'SYMBOL')))[,1])
## 'select()' returned 1:1 mapping between keys and columns
all <- unique(c(HUR_PTBP1$gene_name, HUR$gene_name, PTBP1$gene_name))
entrez_IDS_all <- na.omit(as.data.frame(unlist(mapIds(org.Hs.eg.db, all, 'ENTREZID', 'SYMBOL')))[,1])
## 'select()' returned 1:1 mapping between keys and columns
```

```

BPs <- enrichGO(
  gene = entrez_IDs,
  universe = entrez_IDs_all,
  keyType = "ENTREZID",
  OrgDb = org.Hs.eg.db,
  ont = "BP",
  pAdjustMethod = "BH",
  pvalueCutoff = 0.05,
  qvalueCutoff = 0.05,
  minGSSize = 10,
  readable = TRUE
)

BP <- as.data.frame(BPs)
BP$Description <- factor(BP$Description, levels = BP$Description)

CCs <- enrichGO(
  gene = entrez_IDs,
  universe = entrez_IDs_all,
  keyType = "ENTREZID",
  OrgDb = org.Hs.eg.db,
  ont = "CC",
  pAdjustMethod = "BH",
  pvalueCutoff = 0.05,
  qvalueCutoff = 0.05,
  minGSSize = 10,
  readable = TRUE
)

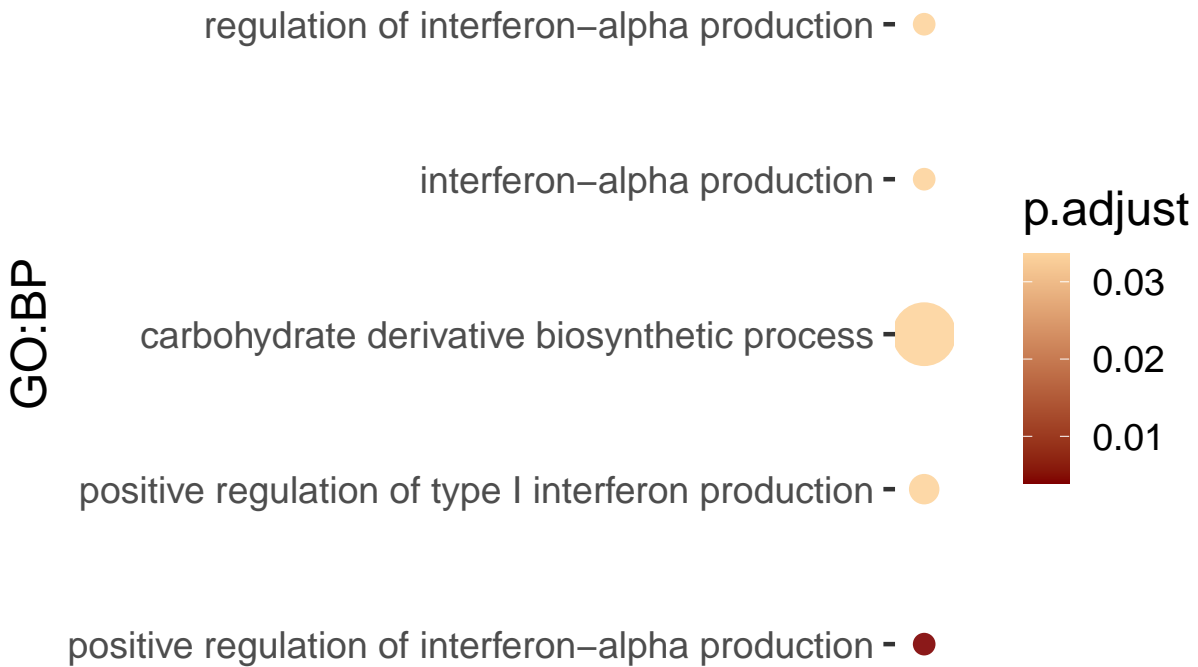
CC <- as.data.frame(CCs)
CC$Description <- factor(CC$Description, levels = CC$Description)

GOs <- rbind(BP,CC)
GO_plot <- ggplot2::ggplot(data=GOs, aes(x=Description, y=1)) +
  geom_point(aes(color=p.adjust, size=sqrt(GOs$Count), alpha=0.9) +
  coord_flip() +
  theme_classic(base_size = 17.5) +
  scale_color_gradient(low="#7f0000",high="#fdd49e") +
  ylab(NULL) +
  # CC is NA
  xlab("GO:BP") +
  theme(axis.text.x = element_blank(),
        axis.line = element_blank(),
        axis.ticks.x = element_blank(),
  )

# write.csv(GOs, "./Data/GOs.csv")

#pdf("./Figure/Figure4/FigureS4F.pdf", width = 9, height = 8)
GO_plot

```

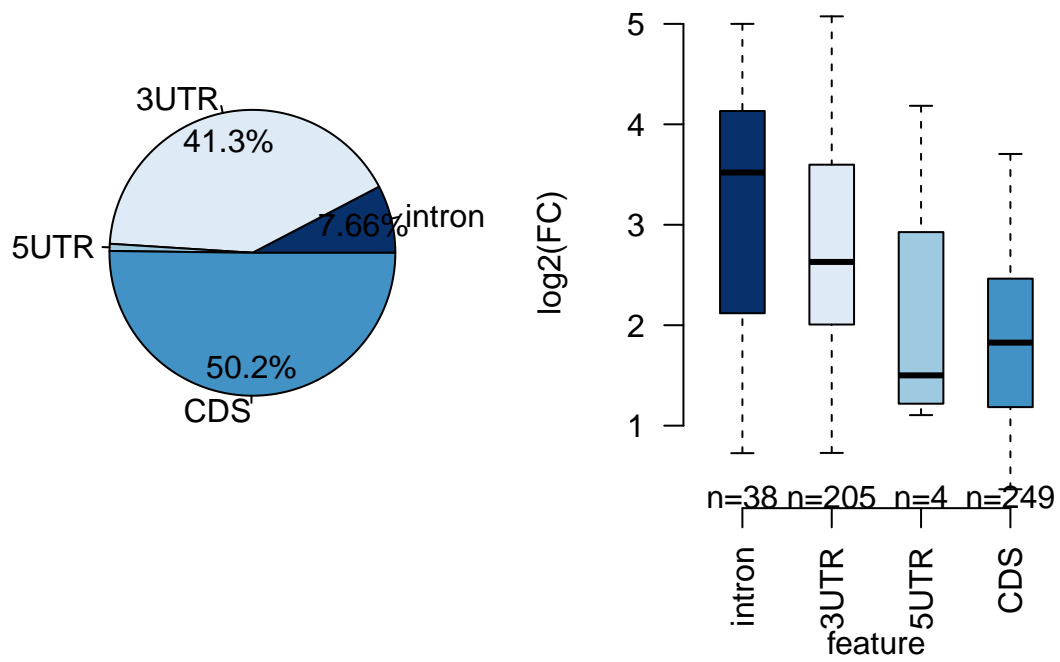



```
#dev.off()
```

0.5.11 Figure S4G

```
gene_name <- unique(unlist(str_split(GOs$geneID,"\\/")))
genes <- as.data.frame(gene_name)

features <- HUR_PTBP1[,c("gene_name", "feature", "BS", "Mean_FCI", "Mean_FCH")]
features <- merge(genes, features, by="gene_name")
features$feature <- factor(features$feature, levels = c("intron", "3UTR", "5UTR", "CDS"))
#pdf("./Figure/Figure4/FigureS4G.pdf", width = 8, height = 5)
par(bty="n", mfrow=c(1,2))
pie1(table(features$feature), percentage=T, col=c("#08306b", "#deebf7", "#9ecae1", "#4292c6"))
boxplot2(data=features, log2(Mean_FCI)~feature, outline=F, col=c("#08306b", "#deebf7", "#9ecae1", "#4292c6"))
```



`#dev.off()`