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 6A . . . 7
0.5.2 Figure 6B . . . 10
0.5.3 Figure 6E . . . 14
0.5.4 Figure 6F . . . 18
0.5.5 Figure S6A . . . 20
0.5.6 Figure S6B . . . 22
0.5.7 Figure S6C . . . 24
0.5.8 Figure S6D . . . 26
0.5.9 Figure S6E right . . . 29
0.5.10 Figure S6F . . . 31
0.5.11 Figure S6G . . . 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 error
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)
```

```r
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", "#8DDBEB", "#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

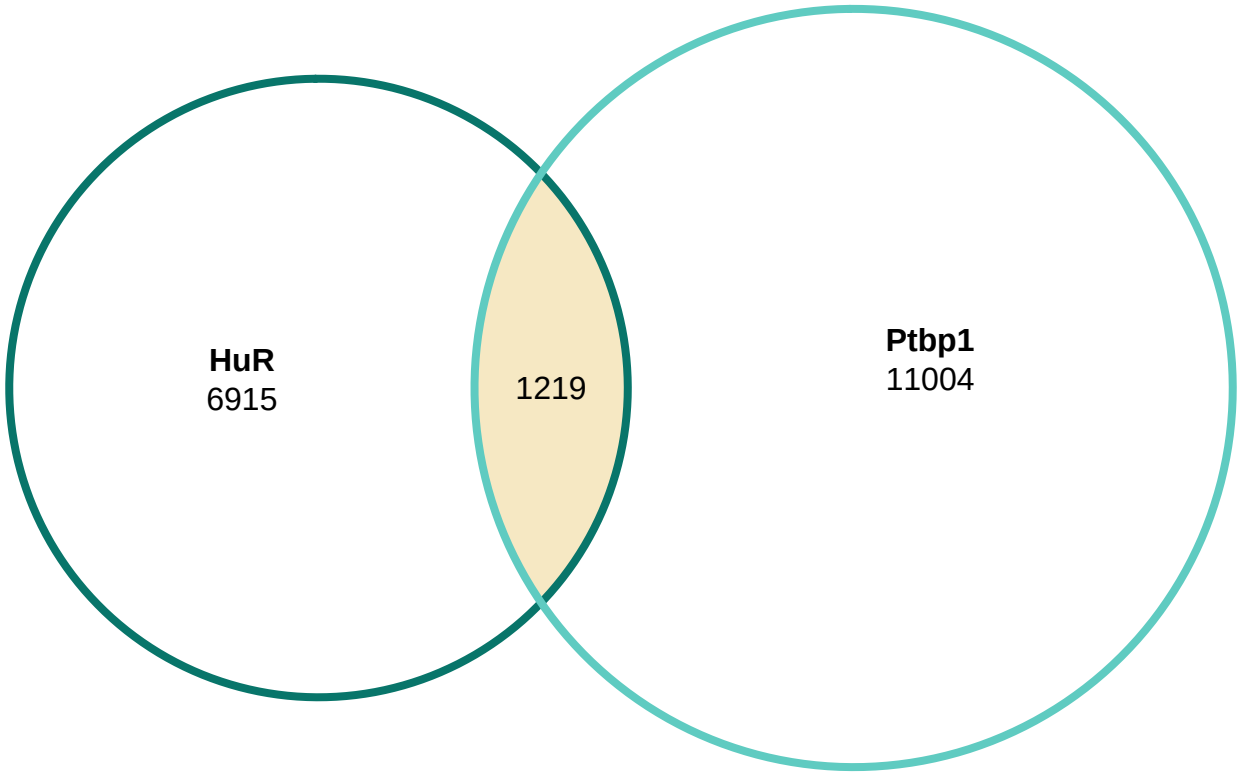
```
# RAP data
HUR <- read.table("./Data/RAPseq/annotated/hsHuRHUMAN.peaks.txt",
                  stringsAsFactors = F, header = T)
PTBP1 <- read.table("./Data/RAPseq/annotated/PTBP1.peaks.txt",
                    stringsAsFactors = F, header = T)
HUR_PTBP1 <- read.table("./Data/RAPseq/annotated/HuRPTBP1.peaks.txt",
                        stringsAsFactors = F, header = T)
print(nrow(HUR))
## [1] 8073
print(nrow(PTBP1))
## [1] 12176
print(nrow(HUR_PTBP1))
## [1] 15029
```

0.5 Figures

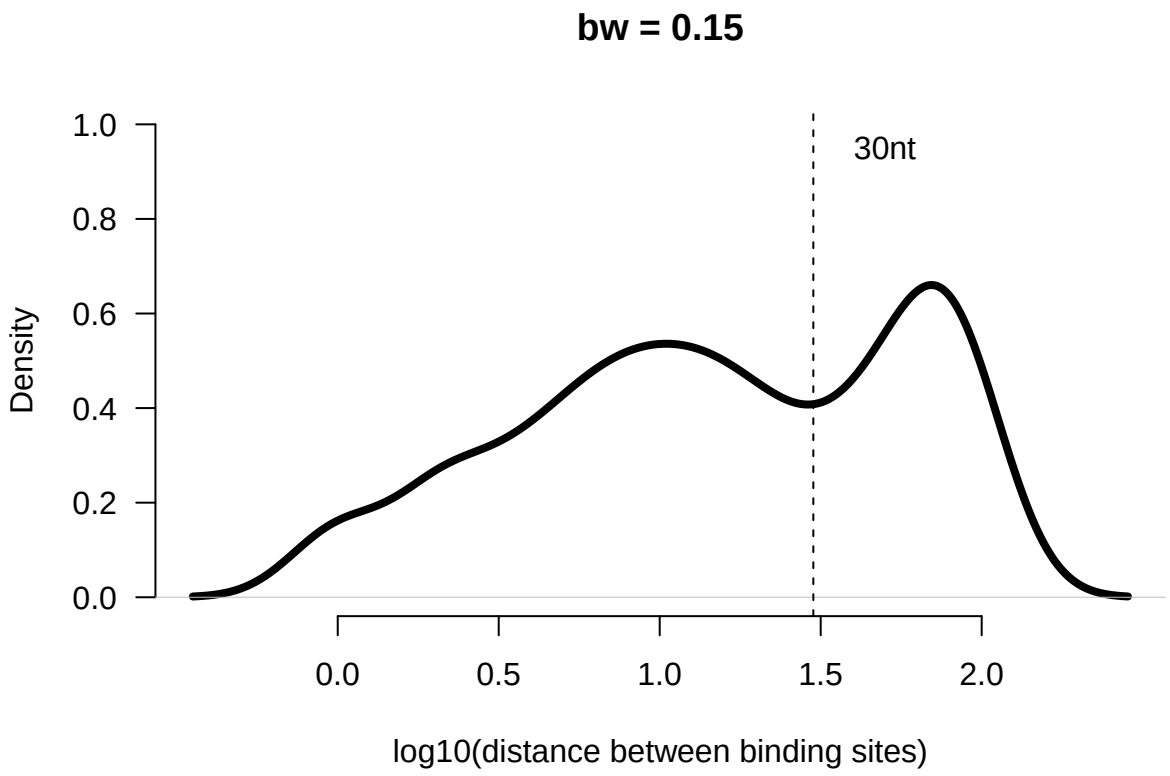
0.5.1 Figure 6A

```
# 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)]
```

```r
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_GRang
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")
```



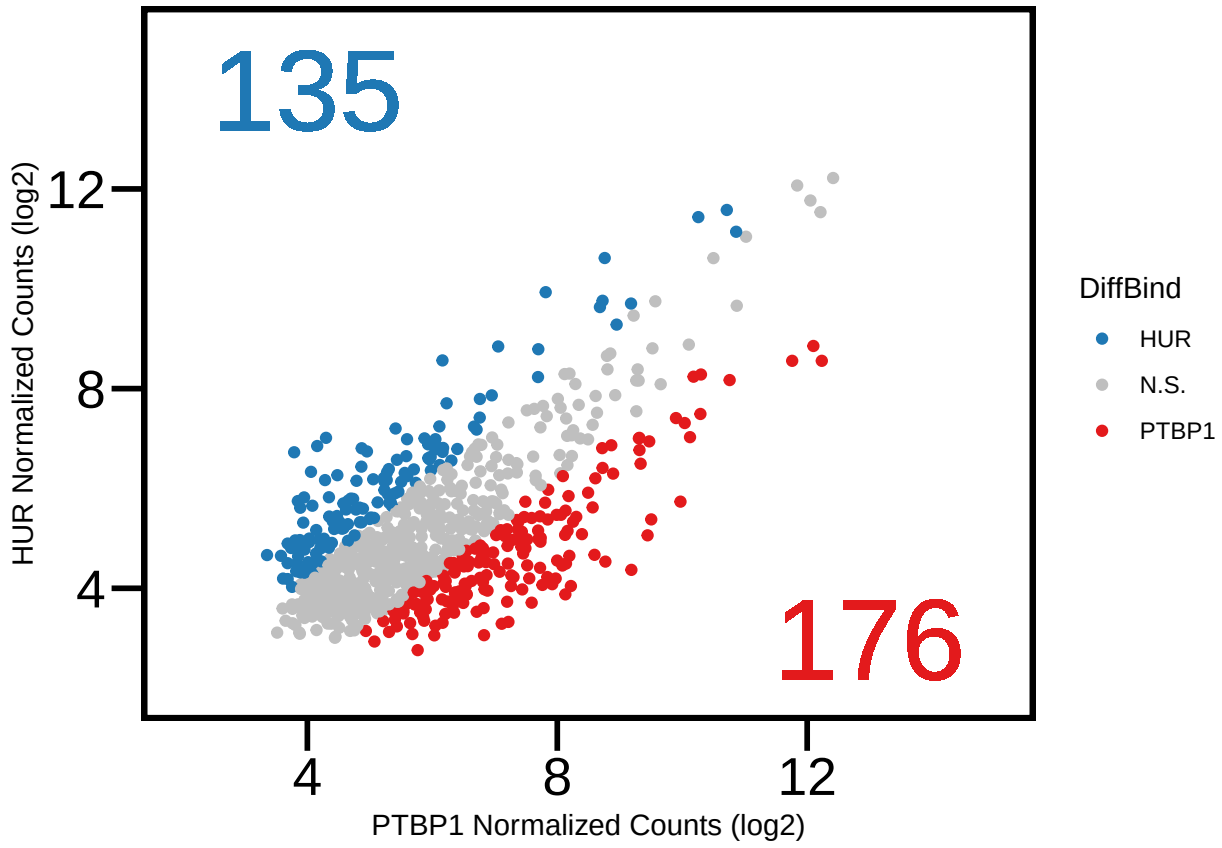

```
#dev.off()
```

0.5.2 Figure 6B

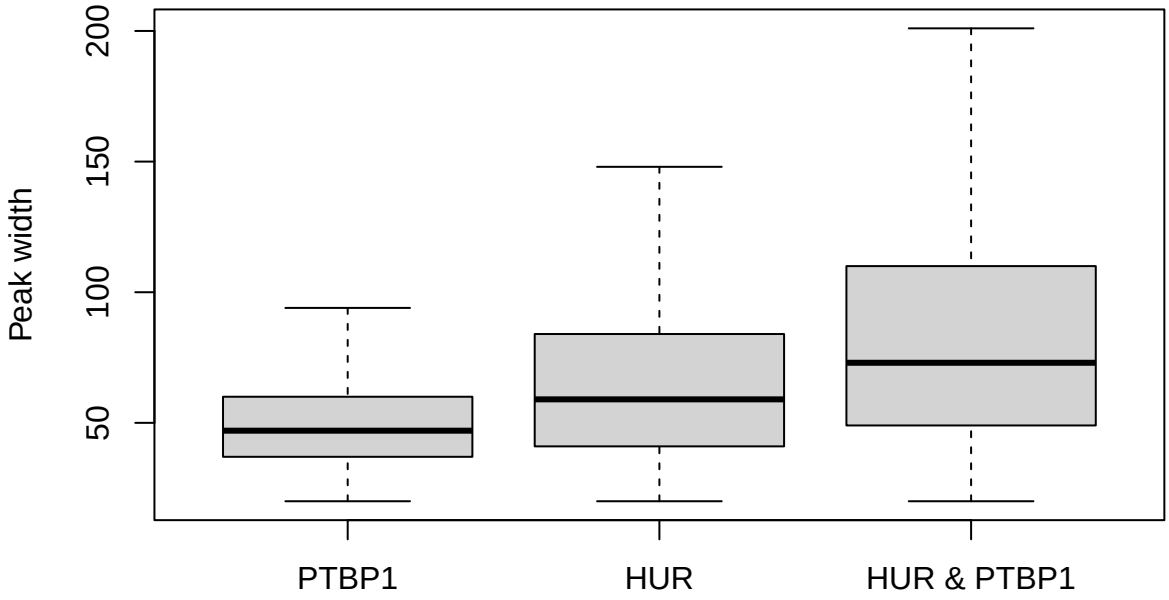
```
# extract 2 replicates
HUR_replicates <- HUR[as.data.frame(findOverlaps(HUR_GRanges, PTBP1_GRanges, type = "any"))[,1],][,c("R
PTBP1_replicates <- PTBP1[as.data.frame(findOverlaps(HUR_GRanges, PTBP1_GRanges, type = "any"))[,2],][,
HUR_comp <- HUR_replicates[log10(abs(PTBP1_overlaps-HUR_overlaps)+1)<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$HUR_rep
HUR_PTBP1_tworeplicates$PTBP1 <- log2((HUR_PTBP1_tworeplicates$PTBP1_rep1 + HUR_PTBP1_tworeplicates$PTB
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$FDR >=
HUR_PTBP1_tworeplicates$PTBP1_won <- HUR_PTBP1_tworeplicates$logFC >= 1 & HUR_PTBP1_tworeplicates$FDR >=
HUR_PTBP1_tworeplicates$DiffBind <- paste(HUR_PTBP1_tworeplicates$HuR_won,HUR_PTBP1_tworeplicates$PTBP1_
HUR_PTBP1_tworeplicates$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$DiffBind)["PTBP1"], s
  geom_text(aes(x=4,y=14), label = table(HUR_PTBP1_tworeplicates$DiffBind)["HUR"], size=15, color="#1f78
## Warning in geom_text(aes(x = 13, y = 3), label = table(HUR_PTBP1_tworeplicates$DiffBind)["PTBP1"], :
## i Please consider using `annotate()` or provide this layer with data containing
##   a single row.
```

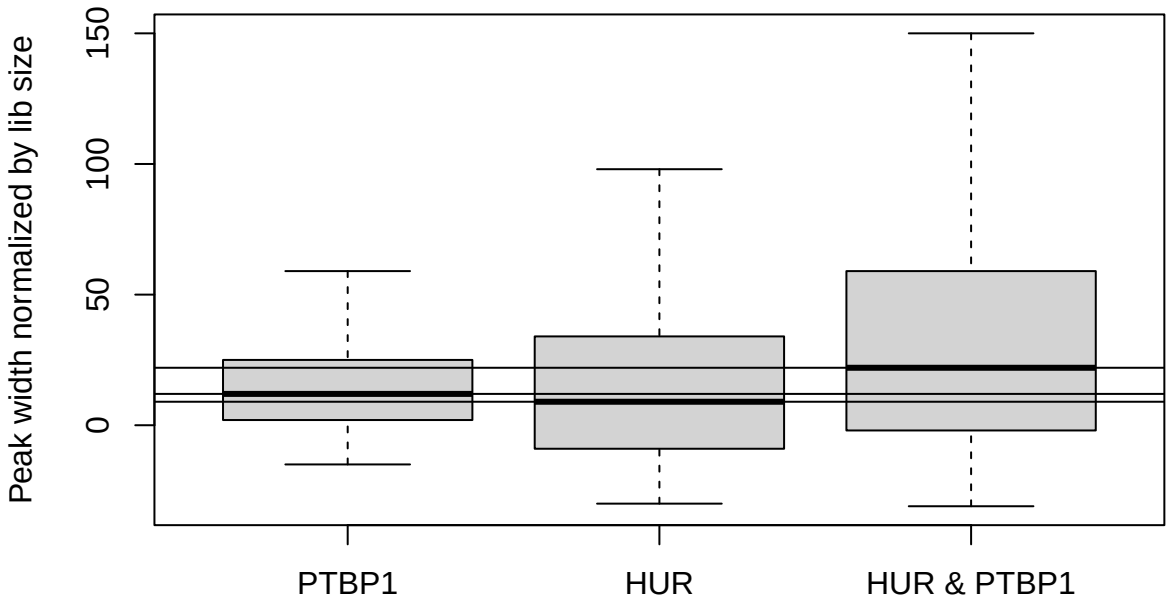
```
## Warning in geom_text(aes(x = 4, y = 14), label = table(HUR_PTBP1_tworeplicates$DiffBind)["HUR"], : A
## i Please consider using `annotate()` or provide this layer with data containing
##   a single row.
```




```
#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")
```



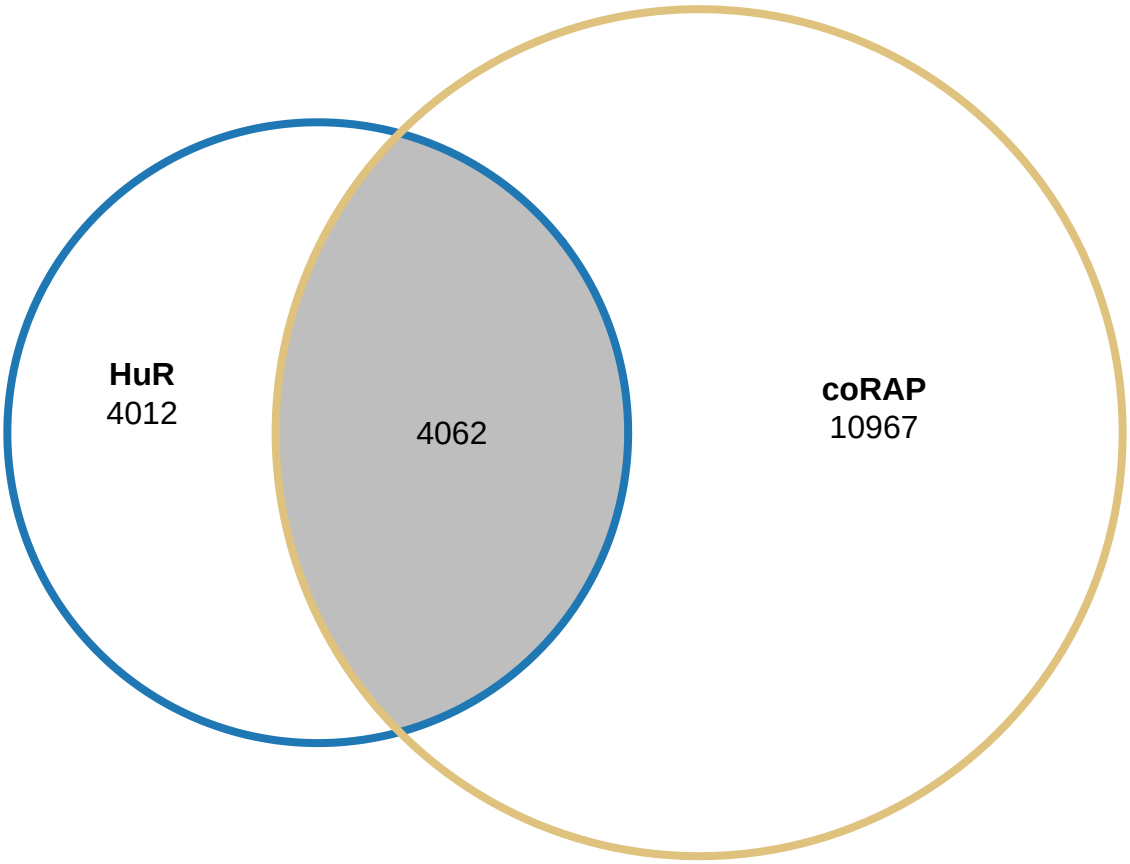

```
boxplot(HUR_PTBP1_len_adjust_list, outline=F, ylab="Peak width normalized by lib size")
abline(h=c(median(PTBP1_len_adjust), median(HUR_len_adjust), median(HUR_PTBP1_len_adjust)))
```



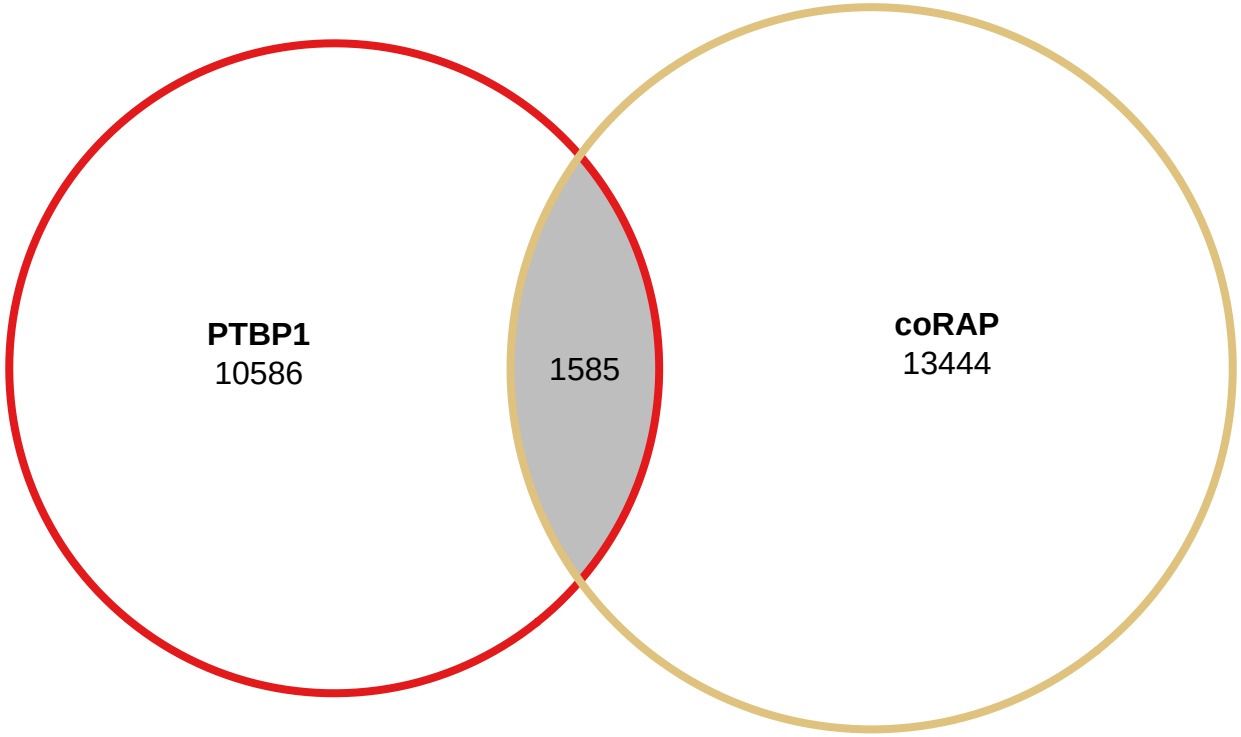
0.5.3 Figure 6E

```
# 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)
```

```r
# 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,typ
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 = "a
PTBP1_HUR_GRanges_uniq <- unique(HUR_PTBP1[as.data.frame(findOverlaps(PTBP1_GRanges,HUR_PTBP1_GRanges,ty
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"))[,1
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
```

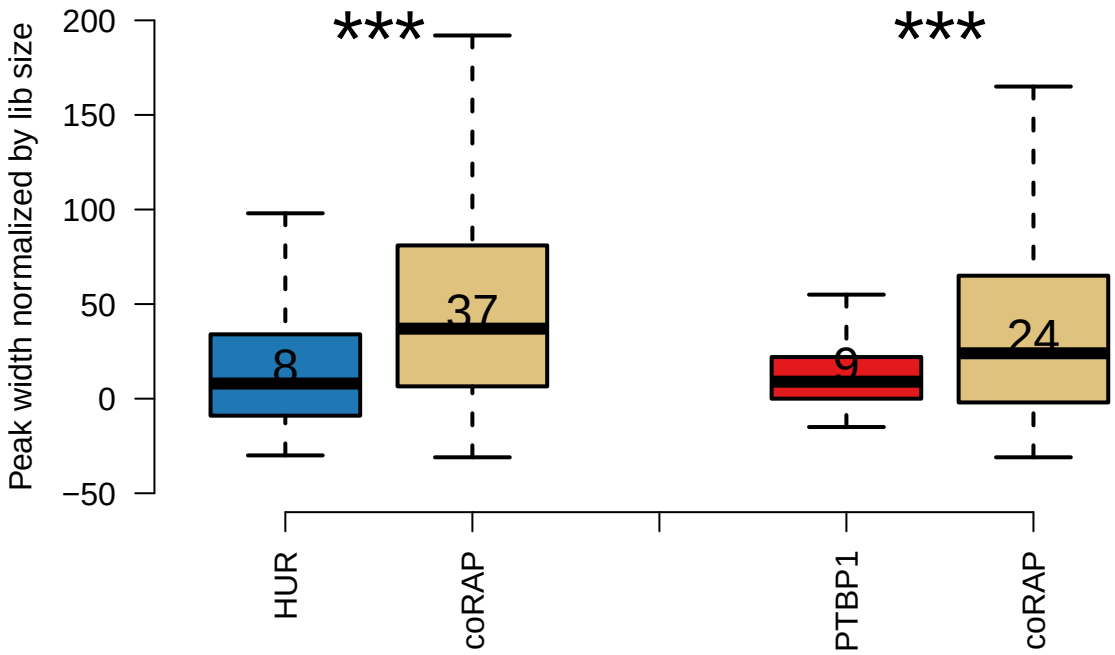


```
#dev.off()

#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("
```




```
#dev.off()

# overlaps
HUR_PTBP1_HURPTBP1_overlaps <- HUR_PTBP1[as.data.frame(findOverlaps(HUR_GRanges,HUR_PTBP1_GRanges,type=
HUR_PTBP1_PTBP1HUR_overlaps <- HUR_PTBP1[as.data.frame(findOverlaps(PTBP1_GRanges,HUR_PTBP1_GRanges,typ

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$end-HUR_PTBP1_PTBP1HUR_overlaps$star
HUR_HURPTBP1_overlaps_adjust <- (HUR_HURPTBP1_overlaps$end - HUR_HURPTBP1_overlaps$start) - 50
HUR_PTBP1_HURPTBP1_overlaps_adjust <- (HUR_PTBP1_HURPTBP1_overlaps$end-HUR_PTBP1_HURPTBP1_overlaps$star

All_overlaps_adjust_list <- list(HUR_HURPTBP1_overlaps_adjust,HUR_PTBP1_HURPTBP1_overlaps_adjust,NULL,P
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(-5
```

```
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_adjus
text(x=4,y=median(PTBP1_HURPTBP1_overlaps_adjust)+9, label=median(PTBP1_HURPTBP1_overlaps_adjust), cex=
text(x=5,y=median(HUR_PTBP1_PTBP1HUR_overlaps_adjust)+9, label=median(HUR_PTBP1_PTBP1HUR_overlaps_adjus

pw <- wilcox.test(PTBP1_HURPTBP1_overlaps_adjust, HUR_PTBP1_PTBP1HUR_overlaps_adjust)
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)
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)
```

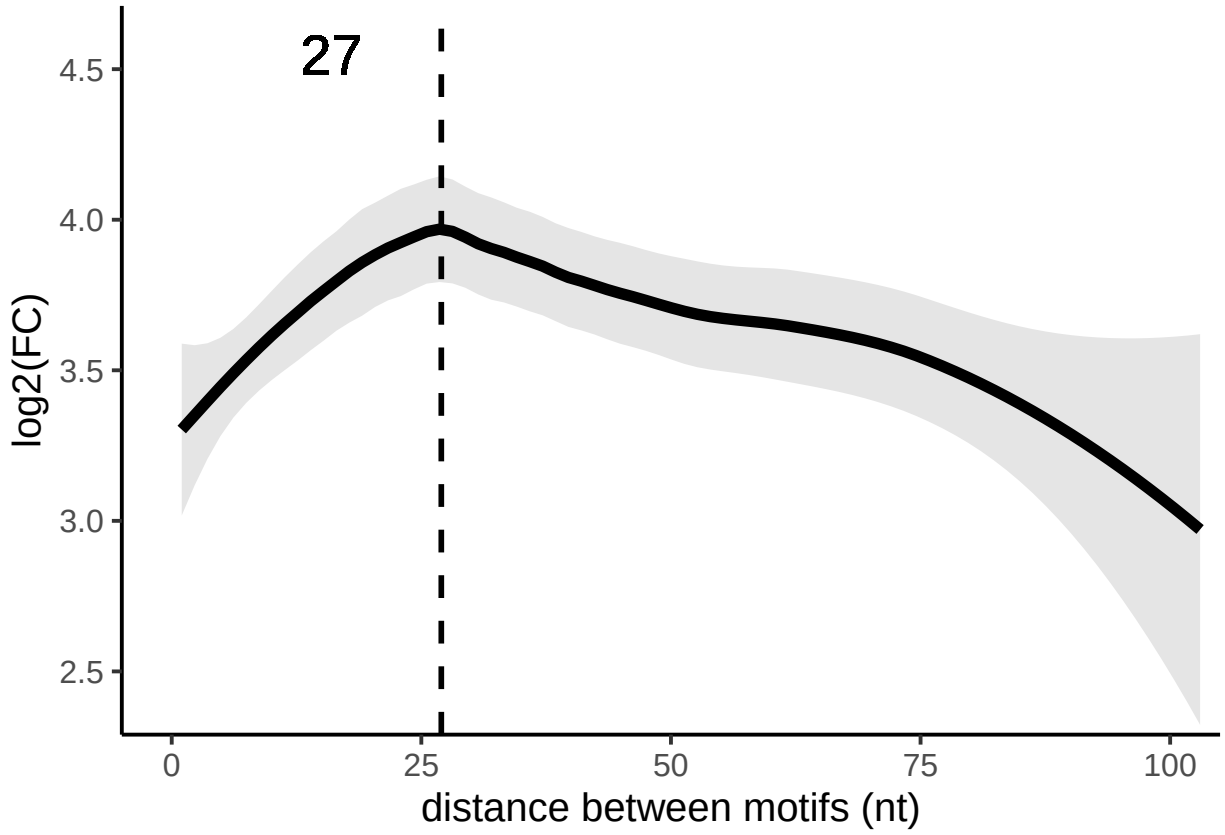



```
#dev.off()
```

0.5.4 Figure 6F

```
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_PTB

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("TTTTTTT","TTTATTT","TTTGTTT","TTTCTTT")
```

```r
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$Ys == 1 & HUR_PTBP
HUR_PTBP1_uniq_4F_overlap_YsUs$Y_loc <- unlist(str_locate_all(HUR_PTBP1_uniq_4F_overlap_YsUs$positive_fa
HUR_PTBP1_uniq_4F_overlap_YsUs$U_loc <- unlist(str_locate_all(HUR_PTBP1_uniq_4F_overlap_YsUs$positive_fa
HUR_PTBP1_uniq_4F_overlap_YsUs$distance <- abs(HUR_PTBP1_uniq_4F_overlap_YsUs$Y_loc - HUR_PTBP1_uniq_4F


HUR_PTBP1_uniq_4F_overlap_YsUs_plot <- ggplot2::ggplot(data=HUR_PTBP1_uniq_4F_overlap_YsUs, aes(x=distan
  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 ha
## i Please consider using `annotate()` or provide this layer with data containing
##   a single row.
## `geom_smooth()` using formula = 'y ~ x'
```

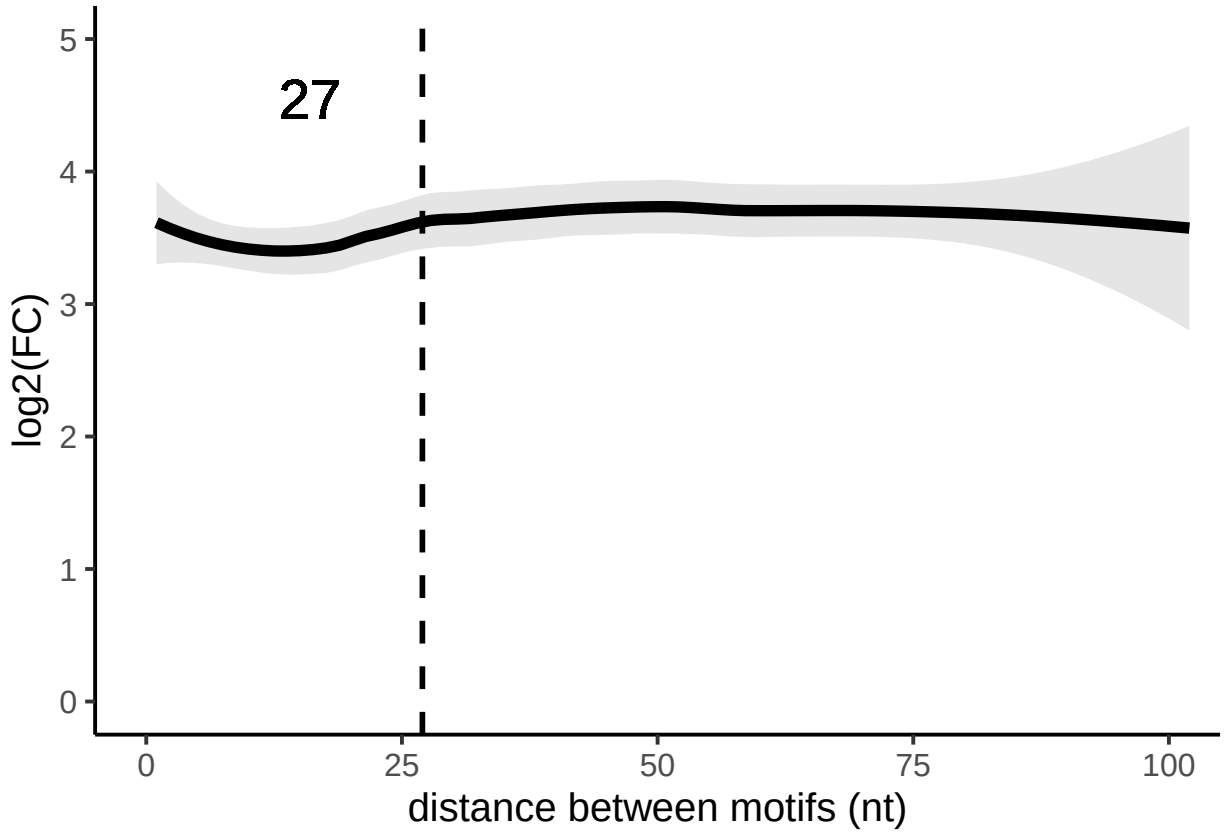



```
#dev.off()
```

0.5.5 Figure S6A

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

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_

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")
```

```r
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$Ys == 1 & HUR_
HUR_PTBP1_uniq_S4A_overlap_YsUs$Y_loc <- unlist(str_locate_all(HUR_PTBP1_uniq_S4A_overlap_YsUs$negative_
HUR_PTBP1_uniq_S4A_overlap_YsUs$U_loc <- unlist(str_locate_all(HUR_PTBP1_uniq_S4A_overlap_YsUs$negative_
HUR_PTBP1_uniq_S4A_overlap_YsUs$distance <- abs(HUR_PTBP1_uniq_S4A_overlap_YsUs$Y_loc - HUR_PTBP1_uniq_


HUR_PTBP1_uniq_S4A_overlap_YsUs_plot <- ggplot2::ggplot(data=HUR_PTBP1_uniq_S4A_overlap_YsUs, aes(x=dis
  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 h
## i Please consider using `annotate()` or provide this layer with data containing
##   a single row.
## `geom_smooth()` using formula = 'y ~ x'
```

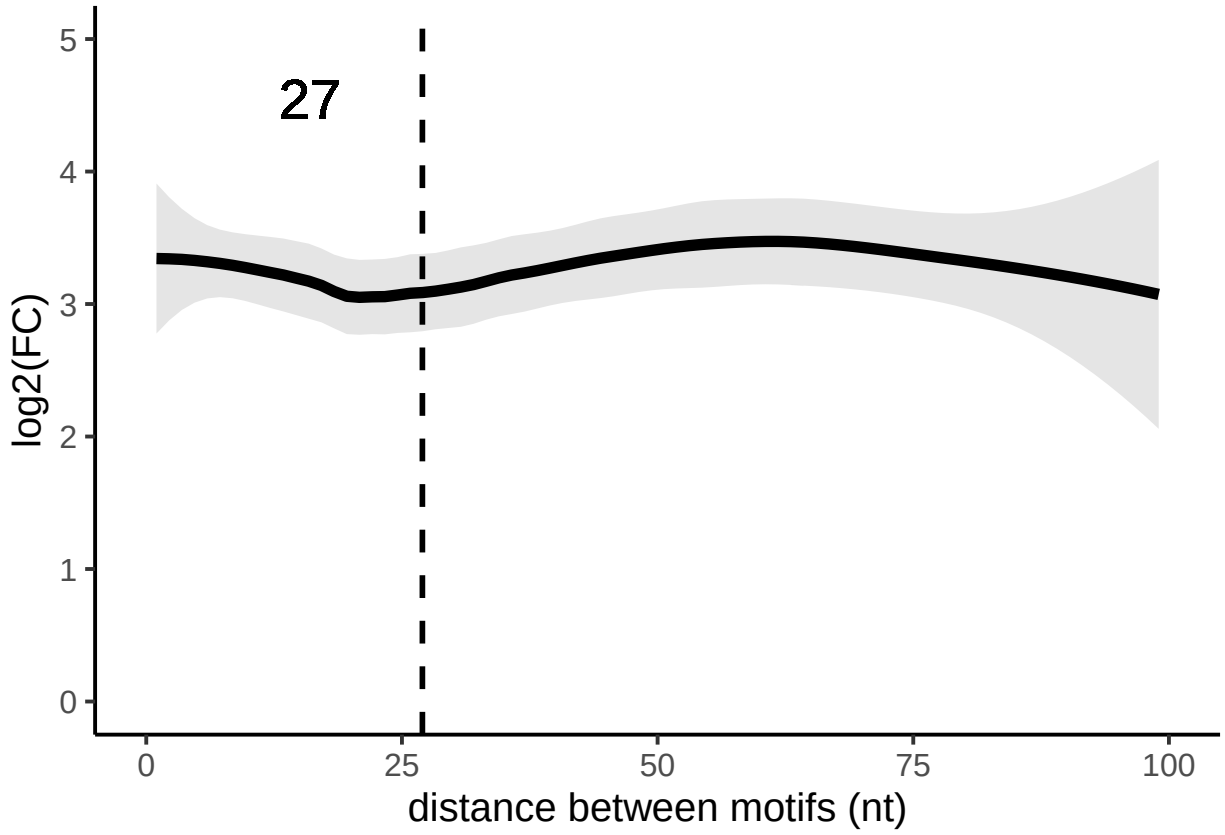



```
#dev.off()
```

0.5.6 Figure S6B

```
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_se

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="")
```

```r
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_uniq_S4B_overlaps$positive_fa <- gsub(i,"U",HUR_uniq_S4B_overlaps$positive_fa)
}

for (i in ptbp1_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$Ys == 1 & HUR_uniq_S4B_overla
HUR_uniq_S4B_overlaps_YsUs$Y_loc <- unlist(str_locate_all(HUR_uniq_S4B_overlaps_YsUs$positive_fa,"Y"))[
HUR_uniq_S4B_overlaps_YsUs$U_loc <- unlist(str_locate_all(HUR_uniq_S4B_overlaps_YsUs$positive_fa,"U"))[
HUR_uniq_S4B_overlaps_YsUs$distance <- abs(HUR_uniq_S4B_overlaps_YsUs$Y_loc - HUR_uniq_S4B_overlaps_YsU


HUR_uniq_S4B_overlaps_YsUs_plot <- ggplot2::ggplot(data=HUR_uniq_S4B_overlaps_YsUs, aes(x=distance, y=l
  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 h
## i Please consider using `annotate()` or provide this layer with data containing
##   a single row.
## `geom_smooth()` using formula = 'y ~ x'
```

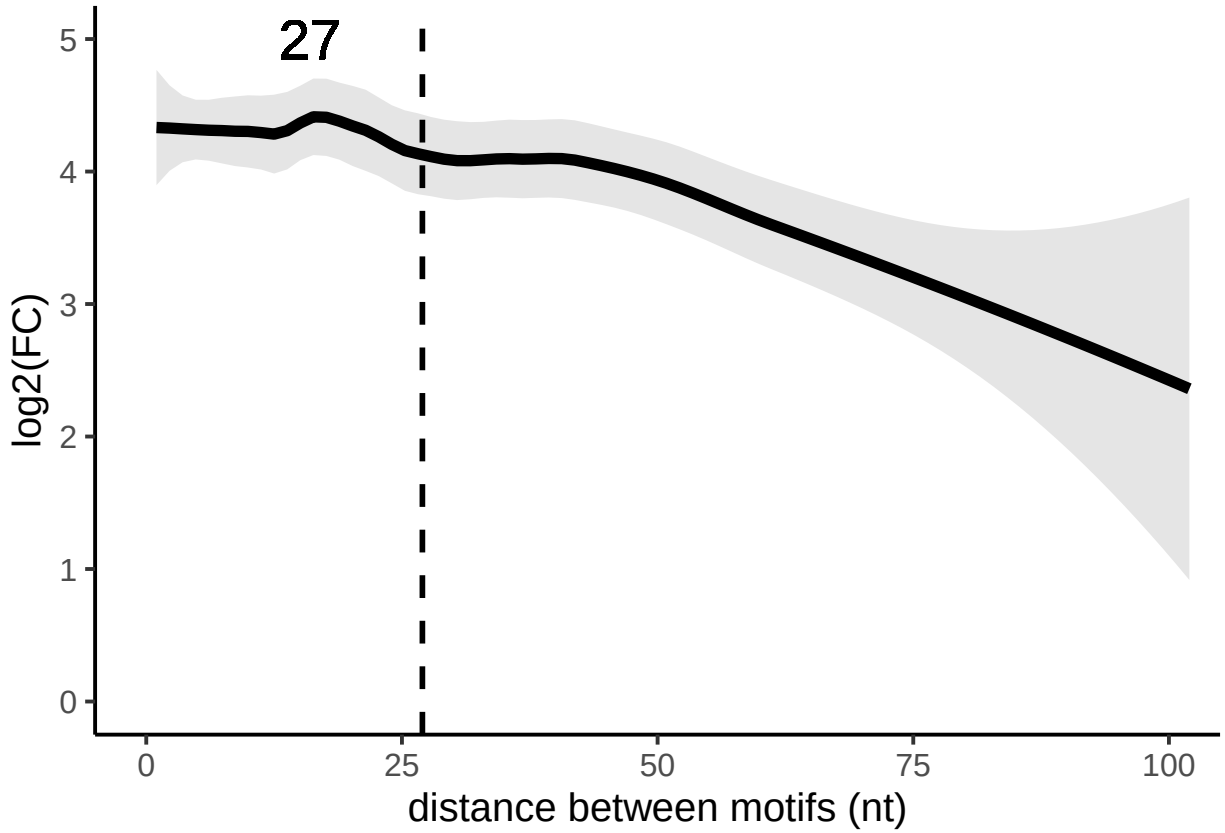



```
#dev.off()
```

0.5.7 Figure S6C

```
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_overlaps$positive_fa <- str_sub(PTBP1_uniq_S4C_select_GRanges_overlaps$pos


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="")
```

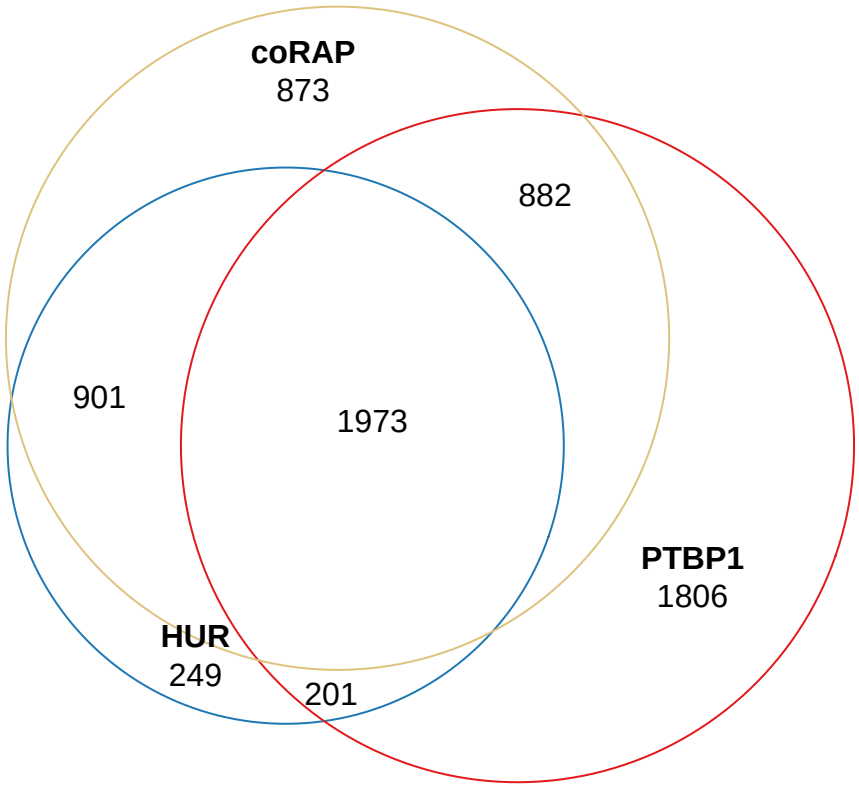
```r
TTT_kmers_1 <- c("TTTTTTT","TTTATTT","TTTGTTT","TTTCTTT")
hur_kmers <- c(TTT_kmers_1,TTT_kmers_2,TTT_kmers_3)
ptbp1_kmers <- c("TCTCT","TTTCT","TTTCT","CTTCT","TCTCT","CCTCT","GTTCT","CTTCT","GCTCT","CCTCT","GTTCT


for (i in hur_kmers){
  PTBP1_uniq_S4C_select_GRanges_overlaps$positive_fa <- gsub(i,"U",PTBP1_uniq_S4C_select_GRanges_overla
}

for (i in ptbp1_kmers){
  PTBP1_uniq_S4C_select_GRanges_overlaps$positive_fa <- gsub(i,"Y",PTBP1_uniq_S4C_select_GRanges_overla
}

PTBP1_uniq_S4C_select_GRanges_overlaps$Ys <- str_count(PTBP1_uniq_S4C_select_GRanges_overlaps$positive_
PTBP1_uniq_S4C_select_GRanges_overlaps$Us <- str_count(PTBP1_uniq_S4C_select_GRanges_overlaps$positive_
PTBP1_uniq_S4C_select_GRanges_overlaps$TTTs <- str_count(PTBP1_uniq_S4C_select_GRanges_overlaps$positive

PTBP1_uniq_S4C_select_GRanges_overlaps_YsUs <- PTBP1_uniq_S4C_select_GRanges_overlaps[PTBP1_uniq_S4C_se
PTBP1_uniq_S4C_select_GRanges_overlaps_YsUs$Y_loc <- unlist(str_locate_all(PTBP1_uniq_S4C_select_GRange
PTBP1_uniq_S4C_select_GRanges_overlaps_YsUs$U_loc <- unlist(str_locate_all(PTBP1_uniq_S4C_select_GRange
PTBP1_uniq_S4C_select_GRanges_overlaps_YsUs$distance <- abs(PTBP1_uniq_S4C_select_GRanges_overlaps_YsUs


PTBP1_uniq_S4C_select_GRanges_overlaps_YsUs_plot <- ggplot2::ggplot(data=PTBP1_uniq_S4C_select_GRanges_
  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 S6D

```
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("#
```

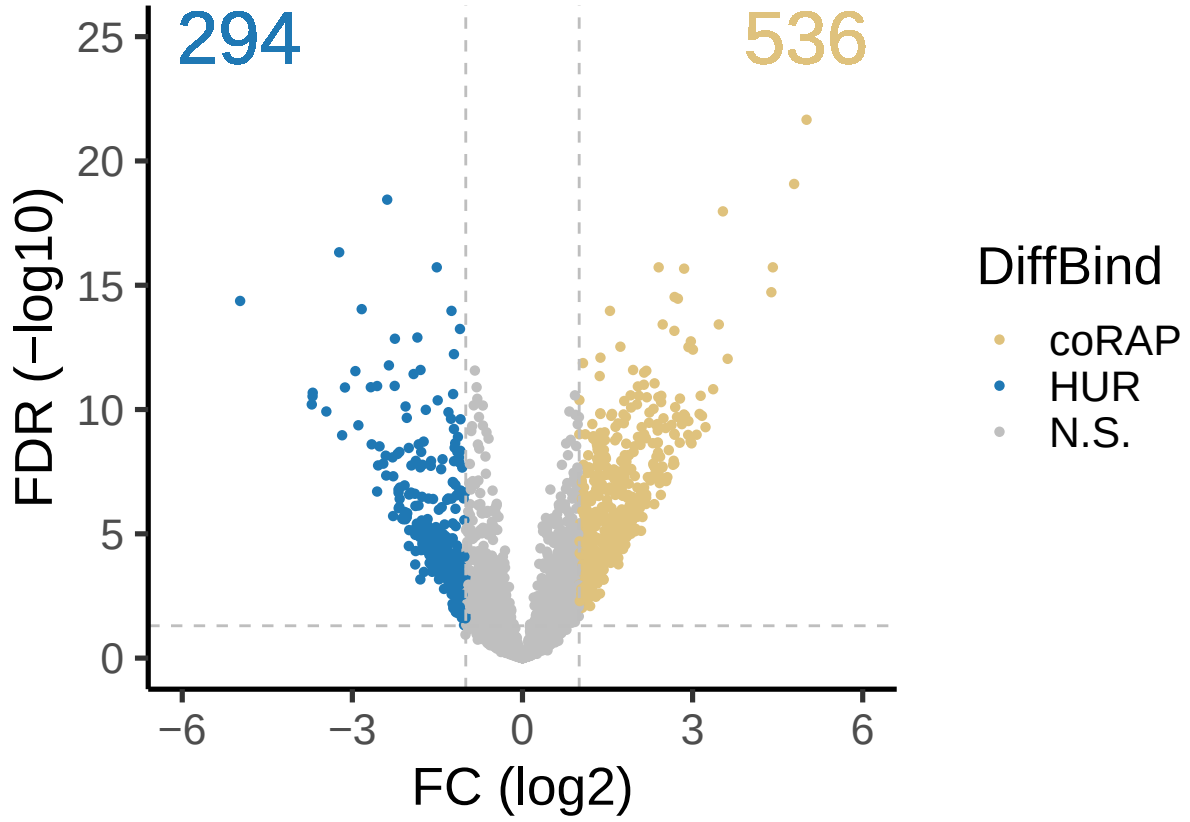



```
#dev.off()
```

###Figure S6E 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 =
## 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(Ge
## 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 >= 1.3010
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_t
  geom_text(aes(x=-5,y=25), label = table(DEG_data_table_S4EL$DiffBind)["HUR"], 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$DiffBind)["coRAP"], : All
## i Please consider using `annotate()` or provide this layer with data containing
##   a single row.
## Warning in geom_text(aes(x = -5, y = 25), label = table(DEG_data_table_S4EL$DiffBind)["HUR"], : All a
## i Please consider using `annotate()` or provide this layer with data containing
##   a single row.
```

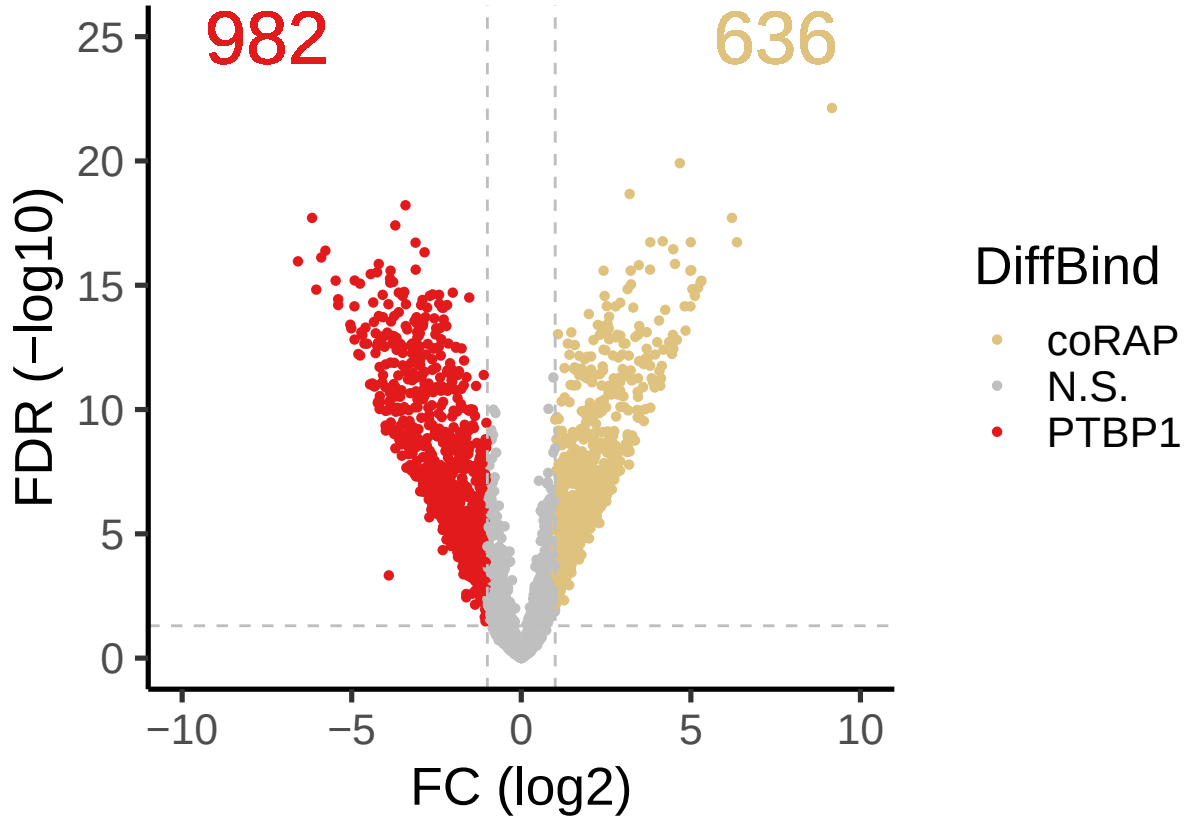



```
#dev.off()
```

0.5.9 Figure S6E 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_Cou
## 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(Ge
## 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 >= 1.3010
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
  geom_text(aes(x=-7.5,y=25), label = table(DEG_data_table_S4ER$DiffBind)["PTBP1"], size=10, color="#e3

#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$DiffBind)["coRAP"], : A
## i Please consider using `annotate()` or provide this layer with data containing
##   a single row.
## Warning in geom_text(aes(x = -7.5, y = 25), label = table(DEG_data_table_S4ER$DiffBind)["PTBP1"], : A
## i Please consider using `annotate()` or provide this layer with data containing
##   a single row.
```

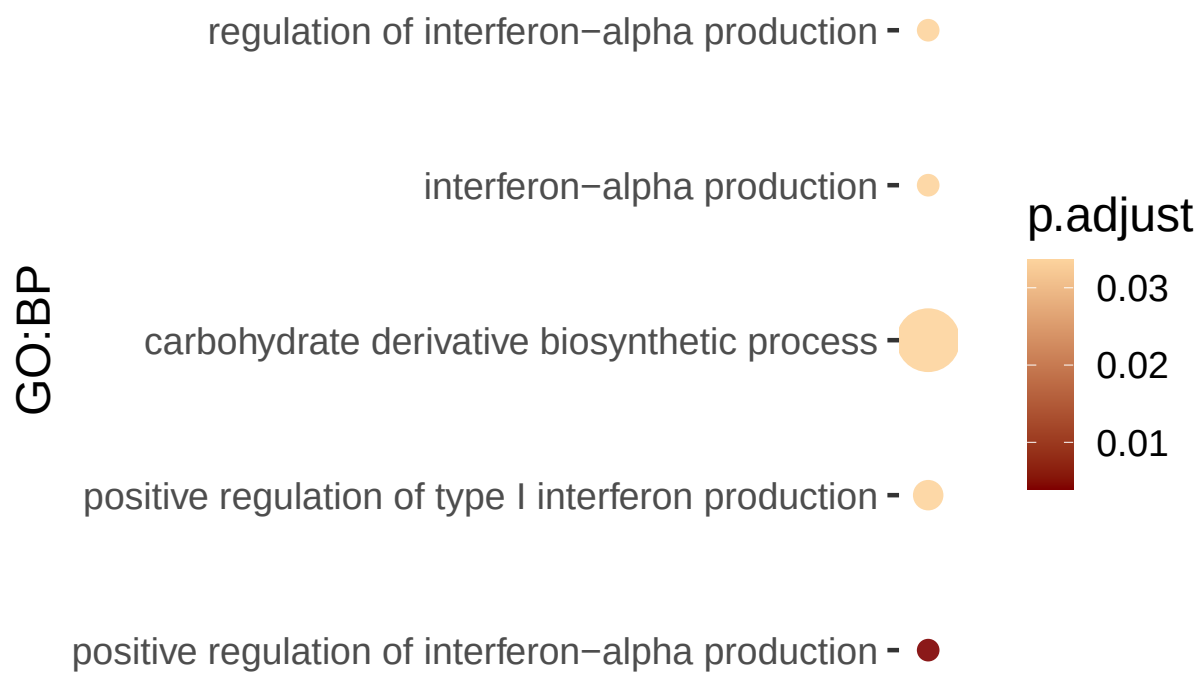



```
#dev.off()
```

0.5.10 Figure S6F

```
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
```

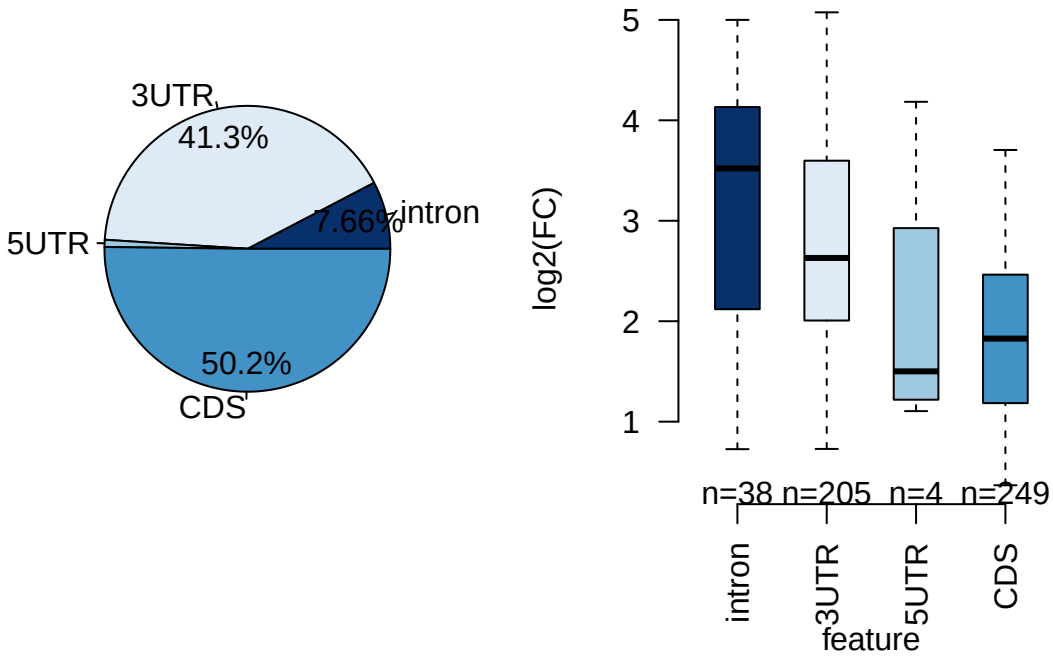
```r
BPs <- enrichGO(
  gene = entrez_IDs,
  universe = entrez_IDs_all,
  keyType       = "ENTREZID",
  OrgDb         = org.Hs.eg.db,
  ont           = "BP",
  pAdjustMethod = "BH",
  pvalueCutoff  = 0.05,
  qvalueCutoff  = 0.05,
  minGSSize     = 10,
  readable      = TRUE
)

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 S6G

```
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", "#4292
```




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

```
sessionInfo()
## R version 4.3.2 (2023-10-31)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS 15.5
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib;  LAPACK v
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: Europe/Stockholm
## tzcode source: internal
##
## attached base packages:
## [1] grid      stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] rrvgo_1.14.2                robustbase_0.99-4
##  [3] ggseqlogo_0.2               eulerr_7.0.2
##  [5] GGally_2.2.1                idr_1.3
```

```
##  [7] ggforce_0.4.2                    corrplot_0.94
##  [9] phastCons100way.UCSC.hg38_3.7.1 phastCons7way.UCSC.hg38_3.7.1
## [11] GenomicScores_2.14.3             ggpubr_0.6.0
## [13] UpSetR_1.4.0                     gridExtra_2.3
## [15] LSD_4.1-0                        VennDiagram_1.7.3
## [17] futile.logger_1.4.3              ReactomePA_1.46.0
## [19] ggbeeswarm_0.7.2                 ggrepel_0.9.5
## [21] gtools_3.9.5                     org.Dr.eg.db_3.18.0
## [23] org.Hs.eg.db_3.18.0              clusterProfiler_4.10.1
## [25] gplots_3.1.3.1                   GenomicFeatures_1.54.4
## [27] AnnotationDbi_1.64.1             Biobase_2.62.0
## [29] reshape2_1.4.4                   dichromat_2.0-0.1
## [31] dendextend_1.17.1                scales_1.3.0
## [33] lubridate_1.9.3                  forcats_1.0.0
## [35] stringr_1.5.1                    dplyr_1.1.4
## [37] purrr_1.0.2                      readr_2.1.5
## [39] tidyr_1.3.1                      tibble_3.2.1
## [41] tidyverse_2.0.0                  ggfortify_0.4.17
## [43] ggplot2_3.5.1                    edgeR_4.0.16
## [45] limma_3.58.1                     ChIPpeakAnno_3.36.1
## [47] GenomicRanges_1.54.1             GenomeInfoDb_1.38.8
## [49] IRanges_2.36.0                   S4Vectors_0.40.2
## [51] BiocGenerics_0.48.1
##
## loaded via a namespace (and not attached):
##   [1] fs_1.6.4                      matrixStats_1.3.0
##   [3] bitops_1.0-8                  enrichplot_1.22.0
##   [5] HDO.db_0.99.1                 httr_1.4.7
##   [7] RColorBrewer_1.1-3            InteractionSet_1.30.0
##   [9] tools_4.3.2                   backports_1.5.0
##  [11] utf8_1.2.4                    R6_2.5.1
##  [13] HDF5Array_1.30.1              mgcv_1.9-1
##  [15] lazyeval_0.2.2                rhdf5filters_1.14.1
##  [17] withr_3.0.1                   graphite_1.48.0
##  [19] prettyunits_1.2.0             cli_3.6.3
##  [21] formatR_1.14                  scatterpie_0.2.4
##  [23] labeling_0.4.3                slam_0.1-53
##  [25] tm_0.7-14                     askpass_1.2.0
##  [27] Rsamtools_2.18.0              yulab.utils_0.1.7
##  [29] gson_0.1.0                    DOSE_3.28.2
##  [31] BSgenome_1.70.2               rstudioapi_0.16.0
##  [33] RSQLite_2.3.7                 treemap_2.4-4
##  [35] generics_0.1.3                gridGraphics_0.5-1
##  [37] BiocIO_1.12.0                 car_3.1-2
##  [39] GO.db_3.18.0                  Matrix_1.6-5
##  [41] fansi_1.0.6                   abind_1.4-5
##  [43] lifecycle_1.0.4               yaml_2.3.10
##  [45] carData_3.0-5                 SummarizedExperiment_1.32.0
##  [47] rhdf5_2.46.1                  qvalue_2.34.0
##  [49] SparseArray_1.2.4             BiocFileCache_2.10.2
##  [51] blob_1.2.4                    promises_1.3.0
##  [53] crayon_1.5.3                  lattice_0.22-6
##  [55] cowplot_1.1.3                 KEGGREST_1.42.0
```

```
##  [57] pillar_1.9.0                 knitr_1.48
##  [59] fgsea_1.28.0                 rjson_0.2.21
##  [61] codetools_0.2-20             fastmatch_1.1-4
##  [63] glue_1.7.0                   ggfun_0.1.6
##  [65] data.table_1.16.0            vctrs_0.6.5
##  [67] png_0.1-8                    treeio_1.26.0
##  [69] gtable_0.3.5                 cachem_1.1.0
##  [71] xfun_0.47                    mime_0.12
##  [73] S4Arrays_1.2.1               tidygraph_1.3.1
##  [75] survival_3.7-0               pheatmap_1.0.12
##  [77] statmod_1.5.0                interactiveDisplayBase_1.40.0
##  [79] nlme_3.1-166                 ggtree_3.10.1
##  [81] bit64_4.0.5                  progress_1.2.3
##  [83] filelock_1.0.3               vipor_0.4.7
##  [85] KernSmooth_2.23-24           colorspace_2.1-1
##  [87] DBI_1.2.3                    tidyselect_1.2.1
##  [89] bit_4.0.5                    compiler_4.3.2
##  [91] curl_5.2.2                   graph_1.80.0
##  [93] NLP_0.3-0                    xml2_1.3.6
##  [95] DelayedArray_0.28.0          shadowtext_0.1.4
##  [97] rtracklayer_1.62.0           caTools_1.18.2
##  [99] DEoptimR_1.1-3               RBGL_1.78.0
## [101] rappdirs_0.3.3               digest_0.6.37
## [103] rmarkdown_2.28               XVector_0.42.0
## [105] htmltools_0.5.8.1            pkgconfig_2.0.3
## [107] umap_0.2.10.0                MatrixGenerics_1.14.0
## [109] highr_0.11                   dbplyr_2.5.0
## [111] regioneR_1.34.0              fastmap_1.2.0
## [113] rlang_1.1.4                  shiny_1.9.1
## [115] farver_2.1.2                 jsonlite_1.8.8
## [117] BiocParallel_1.36.0          GOSemSim_2.28.1
## [119] RCurl_1.98-1.16              magrittr_2.0.3
## [121] GenomeInfoDbData_1.2.11      ggplotify_0.1.2
## [123] wordcloud_2.6                patchwork_1.2.0
## [125] Rhdf5lib_1.24.2              munsell_0.5.1
## [127] Rcpp_1.0.13                  reticulate_1.37.0
## [129] ape_5.8                      viridis_0.6.5
## [131] stringi_1.8.4                ggraph_2.2.1
## [133] zlibbioc_1.48.2              MASS_7.3-60.0.1
## [135] AnnotationHub_3.10.1         plyr_1.8.9
## [137] ggstats_0.6.0                parallel_4.3.2
## [139] Biostrings_2.70.3            graphlayouts_1.1.1
## [141] splines_4.3.2                multtest_2.58.0
## [143] hms_1.1.3                    polylabelr_0.2.0
## [145] locfit_1.5-9.10              igraph_2.0.3
## [147] ggsignif_0.6.4               biomaRt_2.58.2
## [149] futile.options_1.0.1         BiocVersion_3.18.1
## [151] XML_3.99-0.17                evaluate_0.24.0
## [153] lambda.r_1.2.4               BiocManager_1.30.25
## [155] tzdb_0.4.0                   tweenr_2.0.3
## [157] httpuv_1.6.15                openssl_2.2.1
## [159] polyclip_1.10-7              gridBase_0.4-7
## [161] xtable_1.8-4                 broom_1.0.6
```

```
## [163] restfulr_0.0.15              reactome.db_1.86.2
## [165] RSpectra_0.16-2              tidytree_0.4.6
## [167] rstatix_0.7.2                later_1.3.2
## [169] viridisLite_0.4.2            aplot_0.2.3
## [171] memoise_2.0.1                beeswarm_0.4.0
## [173] GenomicAlignments_1.38.2     timechange_0.3.0
```