Figure3

Riccardo Mosca

2025-05-12

This markdown show how to generate Figure 3 and Figure S3

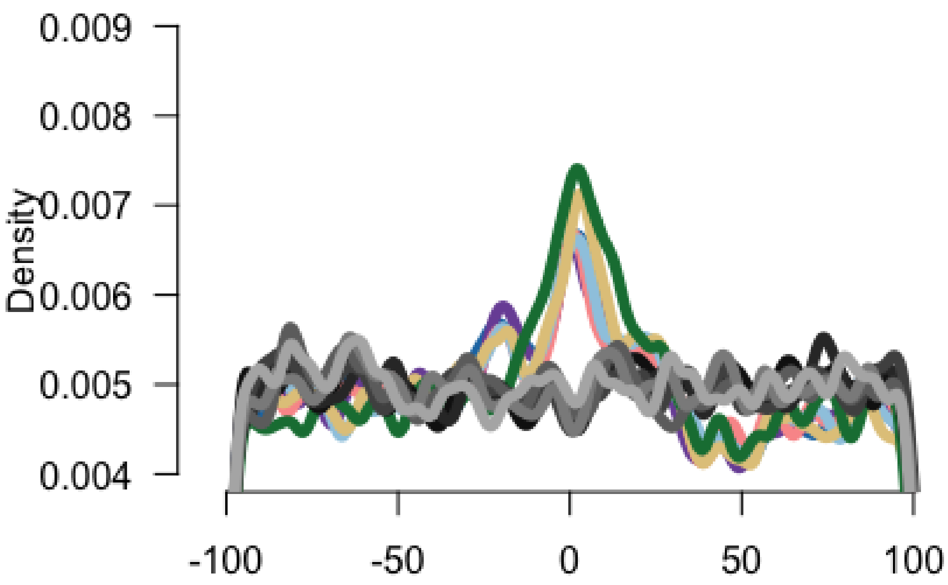
Loading packages and the metafile for the six HUR orthologs with the pyhlop100 for each binding site generated in this script: “HUR_metafile_annotation”

```
library(ggplot2)
library(ggpubr)
library(gplots)
library(stringr)
library(org.Dr.eg.db)
library(clusterProfiler)
library(org.Hs.eg.db)
library(eulerr)

HURs_metafile_annotated <-
read.table("/Users/riccardomosca/Desktop/RAPseq_PAPER/FIGUREs/FIGURE3/HURs_me
tafile_Human_annotated_with_phylop100.txt",
    sep = "\t", header = T)
```

#Figure 3D

```
plot(
density(unlist(str_locate_all(HURs_metafile_annotated[grep("hs",HURs_metafile
_annotated$RBPs),"pos_fa"],"TTT"))-100,bw=2) , lwd=5, col="#1F78B2", bty="n",
main=NA, xlab=NA, ylim=c(0.004,0.009), las=1)
points(
density(unlist(str_locate_all(HURs_metafile_annotated[grep("mm",HURs_metafile
_annotated$RBPs),"pos_fa"],"TTT"))-100,bw=2), type="l", lwd=5, col="#7A52A5")
points(
density(unlist(str_locate_all(HURs_metafile_annotated[grep("md",HURs_metafile
_annotated$RBPs),"pos_fa"],"TTT"))-100,bw=2), type="l", lwd=5, col=
"#FA9999")
points(
density(unlist(str_locate_all(HURs_metafile_annotated[grep("gg",HURs_metafile
_annotated$RBPs),"pos_fa"],"TTT"))-100,bw=2), type="l", lwd=5, col="#9ECAE1")
points(
density(unlist(str_locate_all(HURs_metafile_annotated[grep("xt",HURs_metafile
_annotated$RBPs),"pos_fa"],"TTT"))-100,bw=2), type="l", lwd=5, col=
"#E2C88A")
points(
density(unlist(str_locate_all(HURs_metafile_annotated[grep("dr",HURs_metafile
_annotated$RBPs),"pos_fa"],"TTT"))-100,bw=2), type="l", lwd=5, col="#187D41")
```

```
points(
density(unlist(str_locate_all(HURs_metafile_annotated[grep("hs",HURs_metafile
_annotated$RBPs),"neg_fa"],"TTT"))-100,bw=2), type="l", lwd=5, col= "#141414"
)
points(
density(unlist(str_locate_all(HURs_metafile_annotated[grep("mm",HURs_metafile
_annotated$RBPs),"neg_fa"],"TTT"))-100,bw=2), type="l", lwd=5, col="#323232"
)
points(
density(unlist(str_locate_all(HURs_metafile_annotated[grep("md",HURs_metafile
_annotated$RBPs),"neg_fa"],"TTT"))-100,bw=2), type="l", lwd=5, col="#505050"
)
points(
density(unlist(str_locate_all(HURs_metafile_annotated[grep("gg",HURs_metafile
_annotated$RBPs),"neg_fa"],"TTT"))-100,bw=2), type="l", lwd=5, col="#6E6E6E"
)
points(
density(unlist(str_locate_all(HURs_metafile_annotated[grep("xt",HURs_metafile
_annotated$RBPs),"neg_fa"],"TTT"))-100,bw=2), type="l", lwd=5, col="#8C8C8C"
)
points(
density(unlist(str_locate_all(HURs_metafile_annotated[grep("dr",HURs_metafile
_annotated$RBPs),"neg_fa"],"TTT"))-100,bw=2), type="l", lwd=5, col="#B3B3B3"
)
```



#Figure 3E Plotting a loess smooth curve and boxplots of the ortholog FC normalized to Input across transcripts containing 1–6 consecutive ‘UUU’ motifs.

```
HURs_metafile_annotated$UUUs <-
str_count(str_sub(HURs_metafile_annotated$pos_fa,50,150),"TTT")
HURs_metafile_annotated$UUUs[HURs_metafile_annotated$UUUs > 6] <- 6
RAP <- HURs_metafile_annotated
smooth <- ggplot2::ggplot(data=RAP, aes(x=UUUs, y=Ortholog_average_FCI)) +
  stat_smooth(method="loess", color="#135094", fill="#9ECAE1", span=0.8,
level=0.99) +
  theme_classic2(base_size = 5) +
  ylab("Fold Change") +
  xlab("No of Uracil triplets") +
  scale_x_continuous(breaks = c(0,1,2,3,4,5,6)) +
  coord_cartesian(ylim=c(3.4,7))
boxplot <- ggplot2::ggplot(data=RAP, aes(x=as.character(UUUs),
y=Ortholog_average_FCI)) +
  geom_boxplot(outlier.shape = NA, fill=NA, color="#135094", size=1) +
  theme_classic2(base_size = 5) +
  ylab("Fold Change") +
  xlab("No of Uracil triplets") +
  coord_cartesian(ylim=c(0, 20))
Fig3E <- ggarrange(smooth,boxplot,NULL,ncol=3)
```

```
## `geom_smooth()` using formula = 'y ~ x'

Fig3E
```

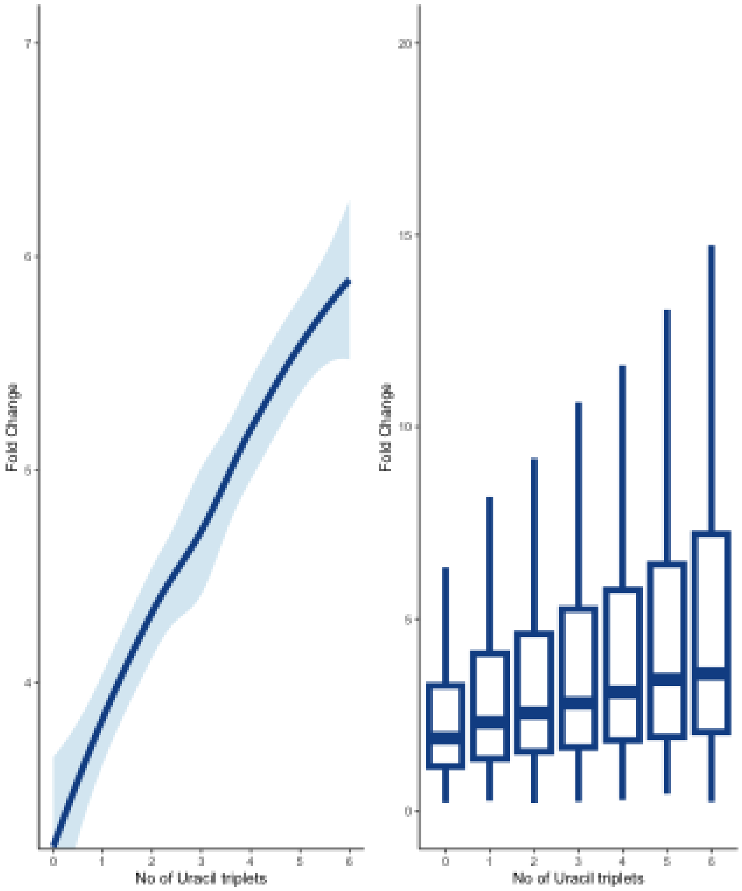
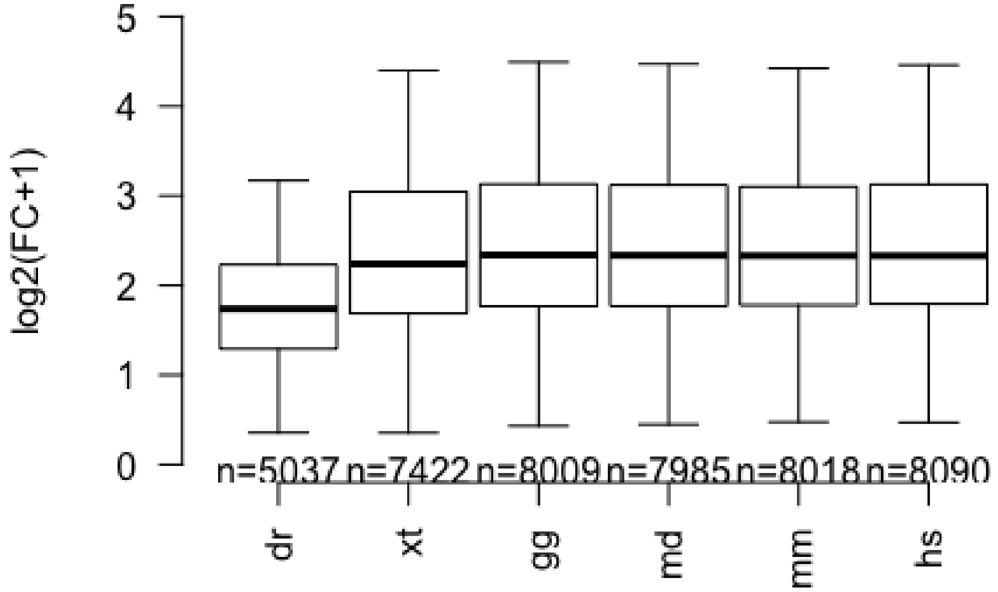


Figure 3F

```
orths <- c("dr","xt","gg", "md", "mm", "hs")
FCIs <- list()
for (i in 1:6){
  FCI <- paste("Mean_FCI_",orths[i],sep="")
  FCH <- paste("Mean_FCH_",orths[i],sep="")
  FCIs[[i]] <-
log2(HURs_metafile_annotated[HURs_metafile_annotated[,FCH]>=2,FCI]+1)

}
names(FCIs) <- orths

par(bty="n")
Fig3F <- boxplot2(FCIs,outline = FALSE, las=2, lty=1, range=1,  col=NA,
ylab="log2(FC+1)", ylim=c(0,5), boxwex=0.9)
```



```r
list <- list(
  dr = HURs_metafile_annotated$Mean_FCI_dr,
  xt  = HURs_metafile_annotated$Mean_FCI_xt
)

# run wilcox.test() for each, comparing to the overlap vector
wt <- lapply(names(list), function(nm) {
  y   <- list[[nm]]
  test <- wilcox.test(HURs_metafile_annotated$Mean_FCI_gg, y)
  data.frame(
    comparison = paste0("gg_vs_", nm),
    W          = as.numeric(test$statistic),
    p.value    = test$p.value
  )
})

# combine into one df
wt_df <- do.call(rbind, wt)

print(wt_df)
```

```
##   comparison         W      p.value
## 1   gg_vs_dr 103403333 0.000000e+00
## 2   gg_vs_xt  81814122 6.048381e-19
```

Figure 3G From this figure, the binding sites without UUUs are removed

```
#filtering step
HURs_metafile_filtered <- HURs_metafile_annotated %>%
  filter(str_count(str_sub(pos_fa, 85, 115), "TTT") > 0)
HURs_metafile_filtered$NumOrthologs <-
as.character(HURs_metafile_filtered$NumOrthologs)
HURs_metafile_filtered$NumOrthologs <-
factor(HURs_metafile_filtered$NumOrthologs, levels =
c("1","2","3","4","5","6"))

#sunsampling to the minimum sample size
balanced_HURs <- HURs_metafile_filtered[1:6, ]
set.seed(10)


for (i in 1:6){
# pull out all rows with exactly k orthologs
  subset_i <- HURs_metafile_filtered[HURs_metafile_filtered$NumOrthologs ==
i,]

# (optional) remove upper outliers in Ortholog_average_BS
  stats_i <- summary(subset_i$Ortholog_average_BS)
  thr_upper <- stats_i[6] + (1.5*(stats_i[6] - stats_i[2]))
  subset_i <- subset_i[subset_i$Ortholog_average_BS <= thr_upper,]

# down-sample to the smallest group size
  sampled_i <- subset_i[sample(
1:nrow(subset_i),min(table(HURs_metafile_filtered$NumOrthologs)) ),]

# append into our growing data.frame
  balanced_HURs <- rbind(balanced_HURs,sampled_i)
}

balanced_HURs <- balanced_HURs[-c(1:6),]



coordinates <- function(x) {
  data.frame(y = median(x, na.rm=T),
             ymin = quantile(x, na.rm=T)[[2]],
             ymax = quantile(x, na.rm=T)[[4]])
}



Fig3G <- ggplot(data=balanced_HURs, aes(x=NumOrthologs,
y=log2(balanced_HURs$Ortholog_average_BS))) +
  geom_violin( bw=0.75) +  # Change the fill color to 'blue'
  stat_summary(fun.data = coordinates, geom = "pointrange", size = 1.5,
na.rm=T) +
  theme_pubr() +
  ylim(2, 10) +
  guides(fill=FALSE)  # This removes the legend for fill

Fig3G
```

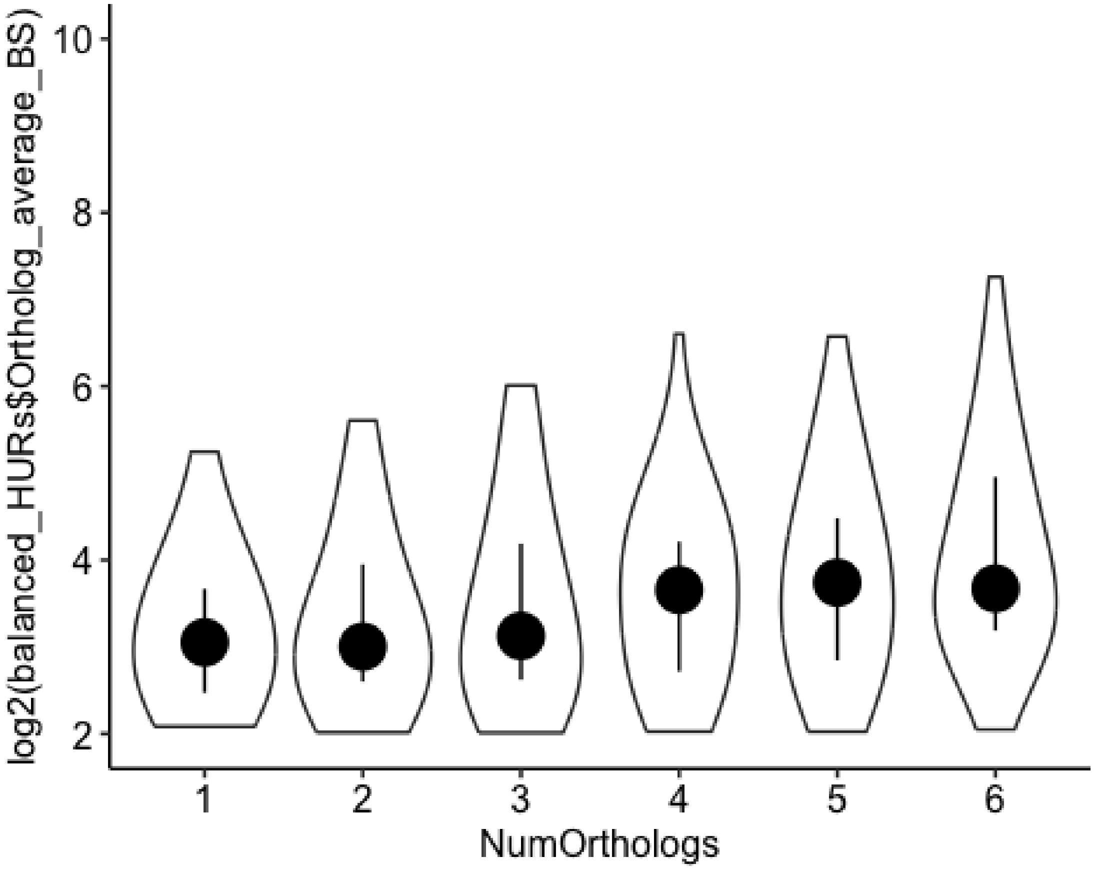


Figure 3H

```
cols <- c("phylop100", "phylop100_neg")

for (col in cols) {
  # replace “.” with NA, then coerce to numeric
  HURs_metafile_filtered[[col]] <- as.numeric(
    replace(
      HURs_metafile_filtered[[col]],
      HURs_metafile_filtered[[col]] == ".",
      NA
    )
  )
}


HURs_metafile_filtered$phylop100 <-
as.numeric(HURs_metafile_filtered$phylop100)
HURs_metafile_filtered$phylop100_neg <-
as.numeric(HURs_metafile_filtered$phylop100_neg)

balanced_HURs <- HURs_metafile_filtered[1:2,]
set.seed(2)
for (i in 1:6){
  subset_i <- HURs_metafile_filtered[HURs_metafile_filtered$NumOrtholog ==
i,]
  sampled_i <- subset_i[sample(
rownames(subset_i),min(table(HURs_metafile_filtered$NumOrtholog)) ),]
  balanced_HURs <- rbind(balanced_HURs,sampled_i)
}
balanced_HURs <- balanced_HURs[-c(1:2),]


coordinates <- function(x) {
  data.frame(y = median(x, na.rm=T),
             ymin = quantile(x, na.rm=T)[[2]],
             ymax = quantile(x, na.rm=T)[[4]])
}

Fig3H <- ggplot(data = balanced_HURs, aes(x=as.character(NumOrthologs)
,y=phylop100, color=as.character(NumOrthologs))) +
  stat_summary(fun.data = coordinates, geom = "pointrange", size = 1.5,
na.rm=T, position = position_nudge(x=-0.1, y=0))  +
  theme_classic() +
  scale_color_manual(values =
c("#135094","#135094","#135094","#135094","#135094", "#135094")) +
  ylab("Mean phyloP 100 vertebrates scores") +
  guides(color=FALSE)

Fig3H + stat_summary(data=balanced_HURs ,
aes(x=as.character(NumOrthologs),y=phylop100_neg), color="#B3B3B3", fun.data
= coordinates, geom = "pointrange", size = 1.5, na.rm=T, position =
position_nudge(x=0.1, y=0))
```

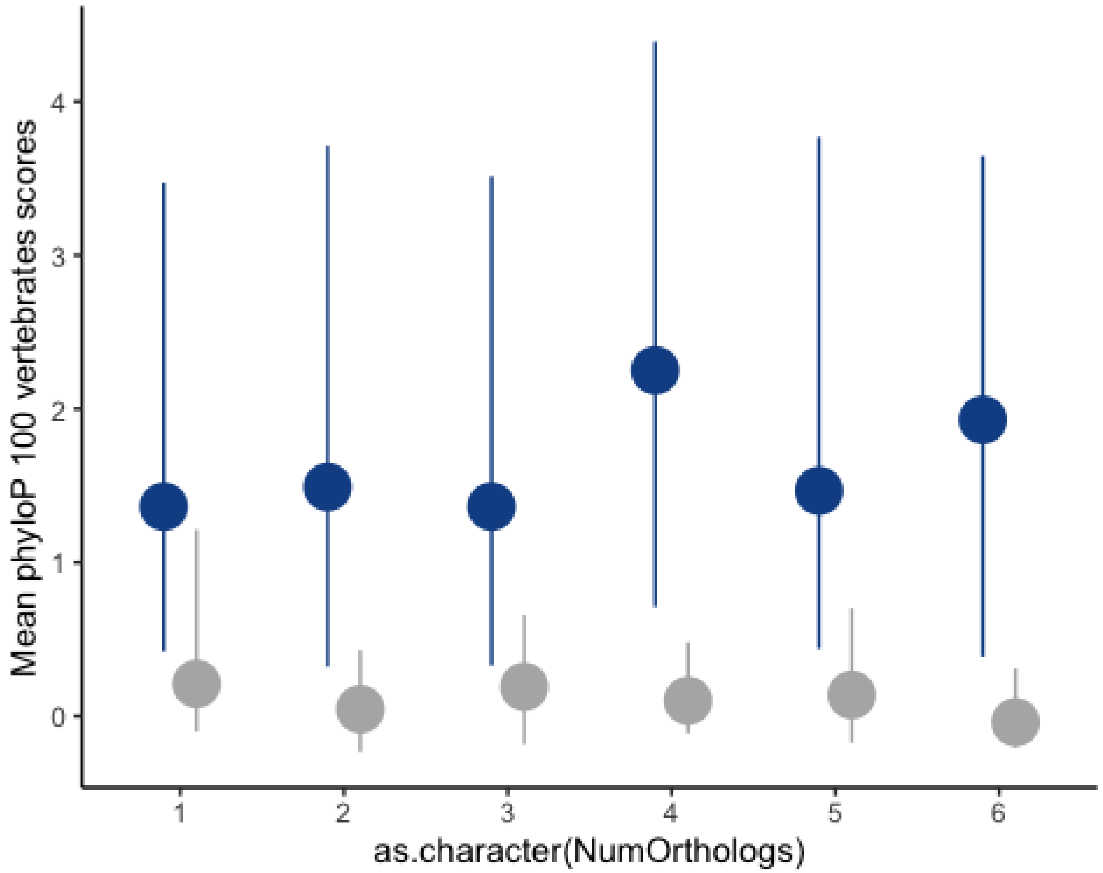


Figure 3I

```
UUUs <- str_count(str_sub(HURs_metafile_filtered$pos_fa,85,115),"TTT")
HURs_metafile_filtered <- HURs_metafile_filtered %>%
  filter(str_count(str_sub(pos_fa, 85, 115), "TTT") > 0) #from 19078 to 12641
UUUs <- str_count(str_sub(HURs_metafile_filtered$pos_fa,85,115),"TTT")
UUUs[UUUs>5] <- 5
UUUs <- factor(as.character(UUUs),levels = c("5","4","3","2","1"))
N_orths <- HURs_metafile_filtered$NumOrthologs
UUUs_table <- table(data.frame(N_orths,UUUs))
UUUs_table <- UUUs_table/rowSums(UUUs_table)

Fig3I <- barplot( t(UUUs_table), col=c("#DEEBF7", "#9ECAE1", "#4292C6",
"#08519C", "#08306B"), space=0.025, las=1 )
```

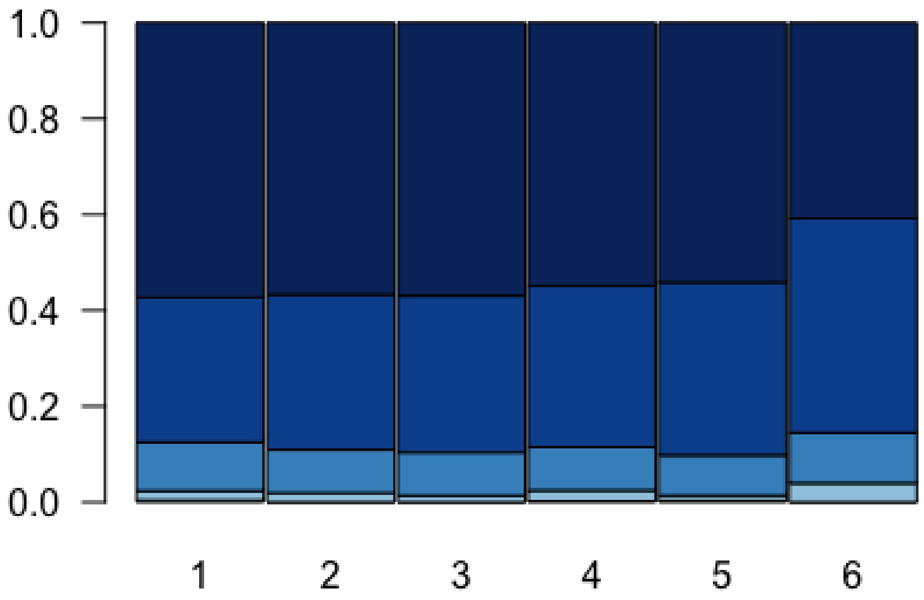
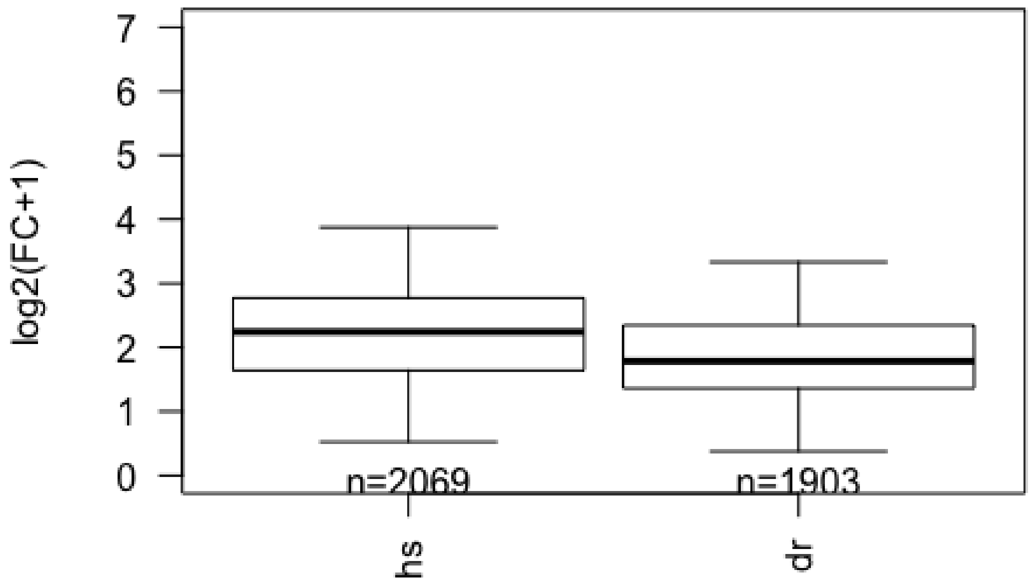


Figure 3J

```
#Loading the metafile with hsHUR and drHUR with the dr substrate
HURs_FISH <- read.table(file =
"/Users/riccardomosca/Desktop/RAPseq_PAPER/FIGUREs/FIGURE3/HURsFISH_metafile_
annotated.txt", header = T, sep="\t")

names(HURs_FISH)[names(HURs_FISH) == "Mean_FCH_hs"] <- "hs_Mean_FCH"
names(HURs_FISH)[names(HURs_FISH) == "Mean_FCI_hs"] <- "hs_Mean_FCI"

names(HURs_FISH)[names(HURs_FISH) == "Mean_FCH_dr"] <- "dr_Mean_FCH"
names(HURs_FISH)[names(HURs_FISH) == "Mean_FCI_dr"] <- "dr_Mean_FCI"


orths <- c("hs","dr")
FCIs <- list()
for (i in 1:2){
  FCI <- paste(orths[i],"_Mean_FCI",sep="")
  FCH <- paste(orths[i],"_Mean_FCH",sep="")
  FCIs[[i]] <- log2(HURs_FISH[HURs_FISH[,FCH]>=2,FCI]+1)

}
names(FCIs) <- orths
```

```
#pdf(file = 
"/Users/ionutatanasoai/Documents/RAPseq_Manuscript/FIGURE_3/HURs_FISH_boxplots_FCs_changes_in_affinity.pdf",3,7.5)
par(mfrow = c(1, 1), mar = c(5.1, 4.1, 4.1, 2.1))
boxplot2(FCIs,outline=F, las=2, lty=1, range=1,  col=NA, ylab="log2(FC+1)", 
ylim=c(0,7), boxwex=0.9)
```



```
#dev.off()

wt <- wilcox.test(HURs_FISH$dr_Mean_FCI, HURs_FISH$hs_Mean_FCI)
wt <- wt$p.value
```

Figure 3K

```
par(mfrow=c(2,2), mar=c(2,2,2,2))
pie(counts_hs_hs,labels = labels_hs_hs, main   = "hs in HUMAN", col=
c("#DEEBF7", "#9ECAE1", "#1F78B2", "#08519C", "#08306B"))
pie(counts_dr_hs,labels = labels_dr_hs, main   = "dr in HUMAN", col=
c("#DEEBF7", "#9ECAE1", "#1F78B2", "#08519C", "#08306B"))
pie(counts_dr_dr,labels = labels_dr_dr, main   = "dr in FISH", col=
c("#DEEBF7", "#9ECAE1", "#1F78B2", "#08519C", "#08306B"))
```

```
pie(counts_hs_dr,labels = labels_hs_dr, main   = "hs in FISH", col=
c("#DEEBF7", "#9ECAE1", "#1F78B2", "#08519C", "#08306B"))
```

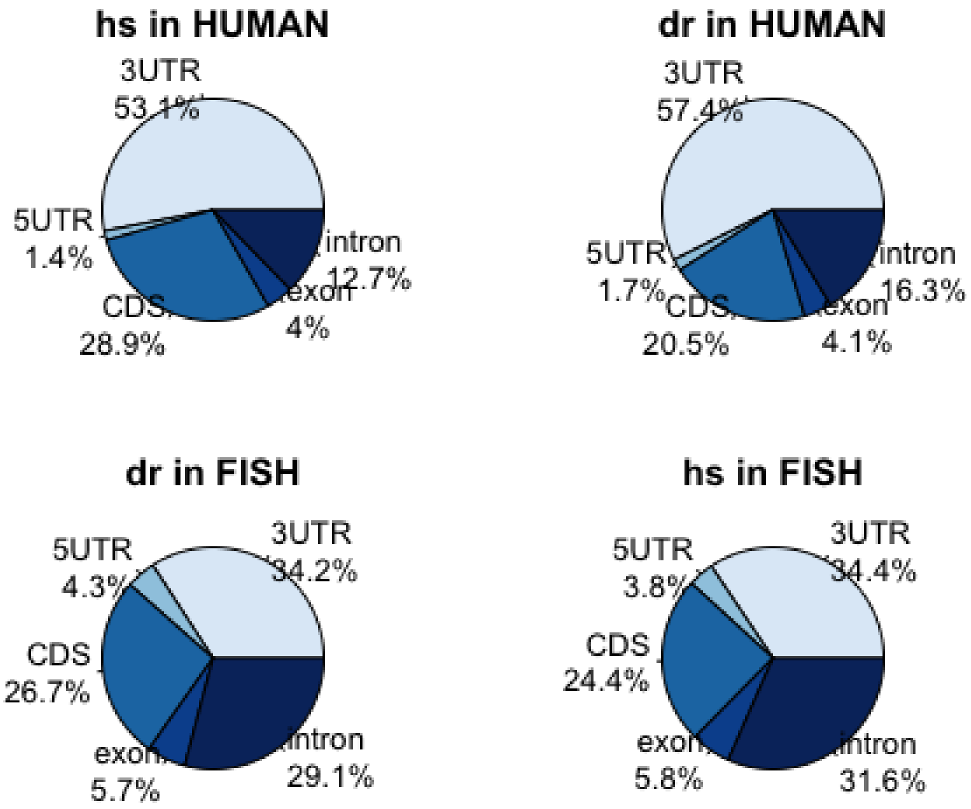


Figure S3C

```
par(mfrow = c(1, 1), mar = c(5.1, 4.1, 4.1, 2.1))



plot(
density(unlist(str_locate_all(HURs_FISH[grep("hsHURFISH",HURs_FISH$RBP),"pos_
fa"],"TTT"))-100,bw=3) , lwd=5, col="#1F78B3", bty="n", main=NA, xlab=NA,
ylim=c(0.003,0.01), las=1)
points(
density(unlist(str_locate_all(HURs_FISH[grep("drHURFISH",HURs_FISH$RBP),"pos_
fa"],"TTT"))-100,bw=3), type="l", lwd=5, col="#197C41")

points(
density(unlist(str_locate_all(HURs_FISH[grep("hsHURFISH",HURs_FISH$RBP),"neg_
fa"],"TTT"))-100,bw=3), type="l", lwd=5, col="#141414" )
points(
density(unlist(str_locate_all(HURs_FISH[grep("drHURFISH",HURs_FISH$RBP),"neg_
fa"],"TTT"))-100,bw=3), type="l", lwd=5, col="#B3B3B3" )
```

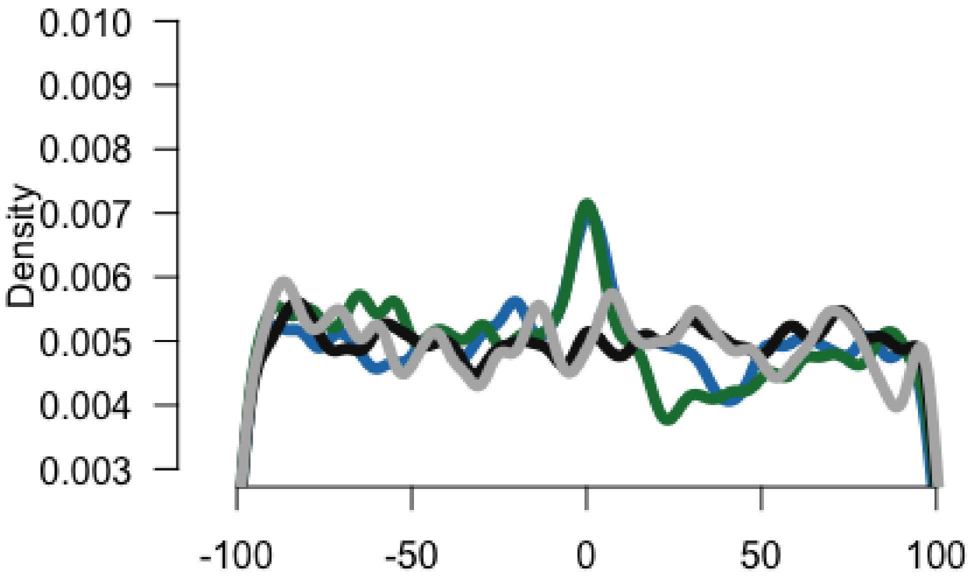


Figure S3D

```
HURs_FISH$UUUs <- str_count(str_sub(HURs_FISH$pos_fa,50,150),"TTT")
HURs_FISH$UUUs[HURs_FISH$UUUs > 7] <- 7
RAP <- HURs_FISH


par(bty="n", mfrow=c(1,2))
boxplot(data=RAP[RAP$BS_hs>=2,], BS_hs~UUUs, outline=F, range=1,
col="#08519C", ylab="hs BS", boxwex=0.9, lty=1)
boxplot(data=RAP[RAP$BS_dr>=2,], BS_dr~UUUs, outline=F, range=1,
col="#197C41", ylab="dr BS", boxwex=0.9, lty=1)
```

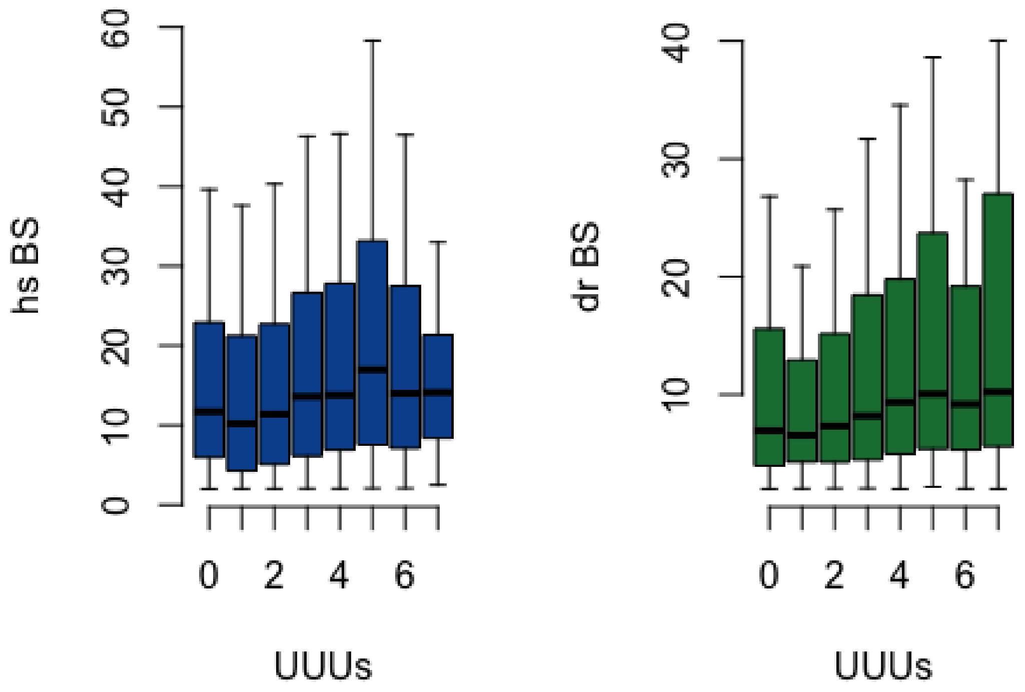


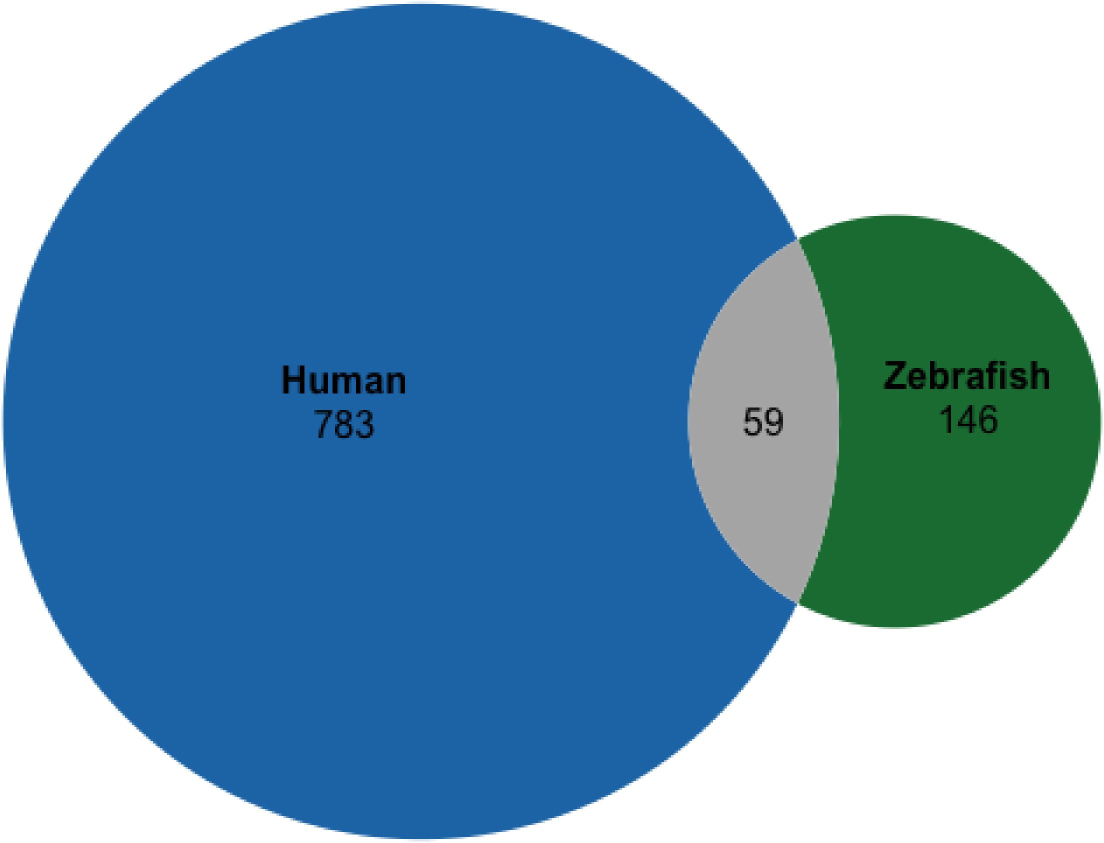
Figure S3E

```
Genes <- unique(HURs_FISH[HURs_FISH$dr_Mean_FCH >= 2 ,"gene_name"])
entrez_IDs <- na.omit(as.data.frame(unlist(mapIds(org.Dr.eg.db, Genes,
'ENTREZID', 'SYMBOL')))[,1])

## 'select()' returned 1:1 mapping between keys and columns

ALL_GOs_dr <- enrichGO(gene = entrez_IDs,
                       keyType       = "ENTREZID",
                       OrgDb         = org.Dr.eg.db,
                       ont           = "ALL",
                       pAdjustMethod = "BH",
                       pvalueCutoff  = 0.01,
                       qvalueCutoff  = 1,
                       minGSSize     = 10,
                       readable      = TRUE)
ALL_GOs_dr <- as.data.frame(ALL_GOs_dr)


Genes <- unique(HURs_metafile_annotated[HURs_metafile_annotated$Mean_FCH_hs
>= 2 ,"gene_name"])
```

```
entrez_IDs <- na.omit(as.data.frame(unlist(mapIds(org.Hs.eg.db, Genes,
'ENTREZID', 'SYMBOL')))[,1])
```

```
## 'select()' returned 1:1 mapping between keys and columns
```

```
ALL_GOs_hs <- enrichGO(gene = entrez_IDs,
                       keyType       = "ENTREZID",
                       OrgDb         = org.Hs.eg.db,
                       ont           = "ALL",
                       pAdjustMethod = "BH",
                       pvalueCutoff  = 0.01,
                       qvalueCutoff  = 1,
                       minGSSize     = 10,
                       readable      = TRUE)
ALL_GOs_hs <- as.data.frame(ALL_GOs_hs)

#Upper one
ABC <- list(unique(ALL_GOs_hs$Description), unique(ALL_GOs_dr$Description))
names(ABC) <- c("Human", "Zebrafish")



FigS3Eup <- euler(ABC, shape="ellipse")
plot(FigS3Eup, fills=c("#1F78B3", "#197C41", "#B3B3B3"), quantities=TRUE,
edges=F)
```



```
#Down one
#Filtering only the common GO terms, and plot the Features distribution
common <- paste(intersect(unique(ALL_GOs_hs$ID),
unique(ALL_GOs_dr$ID)),collapse = "|")

gene_name <-
unique(unlist(str_split(paste(ALL_GOs_hs[grep(common,ALL_GOs_hs$ID),"geneID"]
,collapse = "/"),"\\/")))
df1 <- as.data.frame(gene_name)
df2 <- HURs_metafile_annotated[HURs_metafile_annotated$Mean_FCH_hs >= 2,]
df3 <- merge(df2[grep("intron|3UTR",df2$feature),
c("gene_name","feature")],df1,by="gene_name")
hs_ThreeUTR_Intron_features <- table(df3$feature)

pct <- round(prop.table(hs_ThreeUTR_Intron_features) * 100, 1)
# build labels with newline
labels_hs <- sprintf("%s\n%.1f%%", names(pct), pct)


gene_name <-
unique(unlist(str_split(paste(ALL_GOs_dr[grep(common,ALL_GOs_dr$ID),"geneID"]
,collapse = "/"),"\\/")))
```

```r
df1 <- as.data.frame(gene_name)
df2 <- HURs_FISH[HURs_FISH$dr_Mean_FCH >= 2,]
df3 <-
merge(df2[grep("intron|3UTR",df2$feature),c("gene_name","feature")],df1,by="gene_name")
dr_ThreeUTR_Intron_features <- table(df3$feature)

pct <- round(prop.table(dr_ThreeUTR_Intron_features) * 100, 1)
# build labels with newline
labels_dr <- sprintf("%s\n%.1f%%", names(pct), pct)



par(mfrow=c(1,2))
pie(hs_ThreeUTR_Intron_features, labels_hs, col = c("#DEEBF7", "#08306B"))
pie(dr_ThreeUTR_Intron_features,labels_dr,  col = c("#DEEBF7", "#08306B"))
```

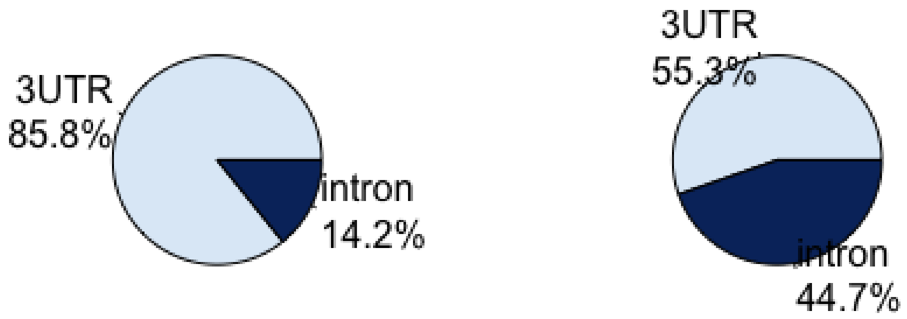


Figure S3F

```r
#extracting gene name per each GO term
Pathways <- ALL_GOs_dr[grep(common,ALL_GOs_dr$ID),c("Description","geneID")]
colnames(Pathways) <- c("Pathway","geneID")
df1 <- data.frame()
for (i in 1:nrow(Pathways)){
  path <- Pathways[i,1]
  genes <- str_split(Pathways[i,2],"\\/")
  path <- rep(path,length(genes))
  gene_GO_match <- data.frame(path,genes)
  colnames(gene_GO_match) <- c("Pathway","geneID")
  df1 <- rbind(df1,gene_GO_match)

}
# each row in df1 correspond to one gene, and the description of the related
GO term
colnames(df1) <- c("Description","gene_name")


Fish_sub <- HURs_FISH[HURs_FISH$dr_Mean_FCH >= 2,]
Fish_sub <- Fish_sub[grep("intron|3UTR",
Fish_sub$feature),c("gene_name","feature")]
dr <- merge(df1,Fish_sub,by="gene_name")
dr <- dr[,c("Description","feature")]
fish_features <- table(dr)/rowSums(table(dr))
fish_features <- fish_features[order(-fish_features[,1]),]

Pathways <- ALL_GOs_hs[grep(common,ALL_GOs_hs$ID),c("Description","geneID")]
colnames(Pathways) <- c("Pathway","geneID")
df1 <- data.frame()
for (i in 1:nrow(Pathways)){

  path <- Pathways[i,1]
  genes <- str_split(Pathways[i,2],"\\/")
  path <- rep(path,length(genes))
  gene_GO_match <- data.frame(path,genes)
  colnames(gene_GO_match) <- c("Pathway","geneID")
  df1 <- rbind(df1,gene_GO_match)

}

colnames(df1) <- c("Description","gene_name")


Human_sub <- HURs_metafile_annotated[HURs_metafile_annotated$Mean_FCH_hs >=
2,]
Human_sub <- Human_sub[grep("intron|3UTR",
Human_sub$feature),c("gene_name","feature")]
hs <- merge(df1,Human_sub,by="gene_name")
hs <- hs[,c("Description","feature")]
human_features <- table(hs)/rowSums(table(hs))
human_features <- human_features[rownames(fish_features),]
```

```
intron_ratios <- fish_features[,2]/human_features[,2]


#Plotting for each GO term the percentage of the binding frequency in fish
intron versus the log2 ratio of fish intron percentage and human intron
percentage. The top intronic binding site bearing GO
#terms highlighted (green)

par(bty="n")
plot(log2(intron_ratios[intron_ratios<=2^1.75]),
fish_features[intron_ratios<=2^1.75, 2]*100,
     ylim=c(0,100),
     xlim=c(-4.5,4.5),
     cex=3,
     pch=16,
     col=alpha("#B3B3B3"),
     las=1,
     xlab="Pathway Intron Ratios log2(Dr/Hs)",
     ylab="Pathway Intron % (Dr)")
points(log2(intron_ratios[intron_ratios>2^1.75]),
fish_features[intron_ratios>2^1.75, 2]*100,
       cex=3,
       pch=16,
       col=alpha("#197C41"),
       xaxt="n",
       yaxt="n")
```

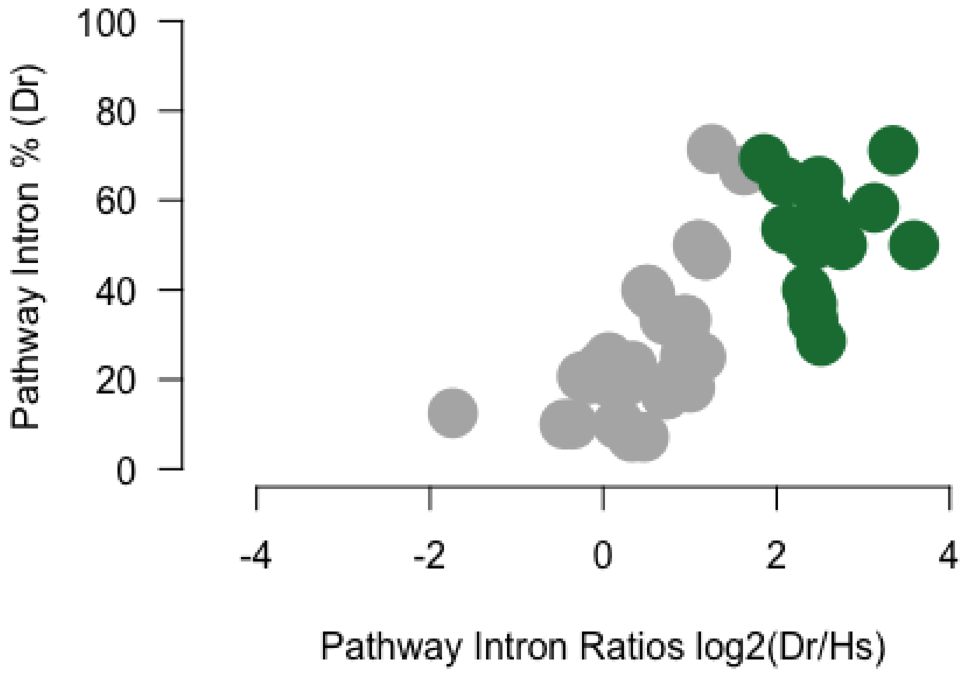
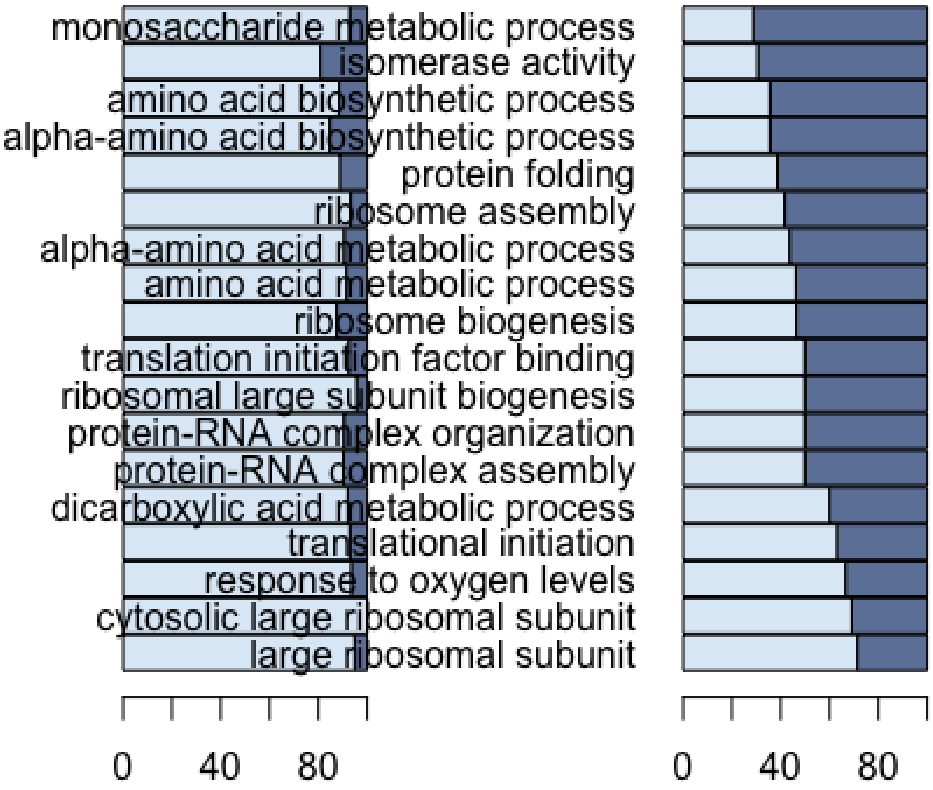


Figure S3G, plotting the GO terms with enriched binding of drHUR in the introninc sites

```
par(mar=c(2,5,2,2), mfrow=c(1,2))
barplot(t(human_features[intron_ratios>2^1.75,])*100, horiz=T, las=1,
col=c("#DEEBF7", "#6E82A7"), space=0.05, yaxt  = "n" )
barplot(t(fish_features[intron_ratios>2^1.75,])*100, horiz=T, las=1,
col=c("#DEEBF7", "#6E82A7"), space=0.05)
```



```
sessionInfo()

## R version 4.3.2 (2023-10-31)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Sonoma 14.5
## 
## 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 version 3.11.0
## 
## 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] stats4    stats     graphics  grDevices utils     datasets  methods  
## [8] base     
## 
## other attached packages:
```

```
##  [1] eulerr_7.0.2          org.Hs.eg.db_3.18.0    clusterProfiler_4.10.1
##  [4] org.Dr.eg.db_3.18.0   AnnotationDbi_1.64.1   IRanges_2.36.0
##  [7] S4Vectors_0.40.2      Biobase_2.62.0         BiocGenerics_0.48.1
## [10] stringr_1.5.1         gplots_3.2.0           ggpubr_0.6.0
## [13] ggplot2_3.5.2
##
## loaded via a namespace (and not attached):
##   [1] RColorBrewer_1.1-3      rstudioapi_0.17.1       jsonlite_2.0.0
##   [4] magrittr_2.0.3          farver_2.1.2            rmarkdown_2.29
##   [7] fs_1.6.6                zlibbioc_1.48.2         vctrs_0.6.5
##  [10] memoise_2.0.1           RCurl_1.98-1.17         ggtree_3.10.1
##  [13] rstatix_0.7.2           htmltools_0.5.8.1       broom_1.0.8
##  [16] Formula_1.2-5           gridGraphics_0.5-1      KernSmooth_2.23-26
##  [19] plyr_1.8.9              cachem_1.1.0            igraph_2.1.4
##  [22] lifecycle_1.0.4         pkgconfig_2.0.3         Matrix_1.6-5
##  [25] R6_2.6.1                fastmap_1.2.0           gson_0.1.0
##  [28] GenomeInfoDbData_1.2.11 digest_0.6.37           aplot_0.2.5
##  [31] enrichplot_1.22.0       colorspace_2.1-1        patchwork_1.3.0
##  [34] RSQLite_2.3.11          labeling_0.4.3          httr_1.4.7
##  [37] polyclip_1.10-7         abind_1.4-8             mgcv_1.9-1
##  [40] compiler_4.3.2          bit64_4.6.0-1           withr_3.0.2
##  [43] backports_1.5.0         BiocParallel_1.36.0     carData_3.0-5
##  [46] viridis_0.6.5           DBI_1.2.3               ggforce_0.4.2
##  [49] ggsignif_0.6.4          MASS_7.3-60.0.1         HDO.db_0.99.1
##  [52] gtools_3.9.5            caTools_1.18.3          tools_4.3.2
##  [55] ape_5.8-1               scatterpie_0.2.4        glue_1.8.0
##  [58] nlme_3.1-168            GOSemSim_2.28.1         polylabelr_0.3.0
##  [61] grid_4.3.2              shadowtext_0.1.4        reshape2_1.4.4
##  [64] fgsea_1.28.0            generics_0.1.3          gtable_0.3.6
##  [67] tidyr_1.3.1             data.table_1.17.0       tidygraph_1.3.1
##  [70] car_3.1-3               XVector_0.42.0          ggrepel_0.9.6
##  [73] pillar_1.10.2           yulab.utils_0.2.0       splines_4.3.2
##  [76] dplyr_1.1.4             tweenr_2.0.3            treeio_1.26.0
##  [79] lattice_0.22-7          bit_4.6.0               tidyselect_1.2.1
##  [82] GO.db_3.18.0            Biostrings_2.70.3       knitr_1.50
##  [85] gridExtra_2.3           xfun_0.52               graphlayouts_1.2.2
##  [88] stringi_1.8.7           lazyeval_0.2.2          ggfun_0.1.8
##  [91] yaml_2.3.10             evaluate_1.0.3          codetools_0.2-20
##  [94] ggraph_2.2.1            tibble_3.2.1            qvalue_2.34.0
##  [97] ggplotify_0.1.2         cli_3.6.5               dichromat_2.0-0.1
## [100] Rcpp_1.0.14             GenomeInfoDb_1.38.8     png_0.1-8
## [103] parallel_4.3.2          blob_1.2.4              DOSE_3.28.2
## [106] bitops_1.0-9            viridisLite_0.4.2       tidytree_0.4.6
## [109] scales_1.4.0            purrr_1.0.4             crayon_1.5.3
## [112] rlang_1.1.6             cowplot_1.1.3           fastmatch_1.1-6
## [115] KEGGREST_1.42.0         formatR_1.14
```