Figure 5

Riccardo Mosca

2025-05-13

Loading packages, and the follwing data: IGF2BP1-2-3 mutations in HCC patients, downloaded from BioMuta on 2020 - 04 - 12 and the IGF2BP1-2-3 metafile generated using the followng script “IGF2BP1_metafile_annotation”

```
library(tidyverse)
library(ggplot2)
library(ggpubr)
library(edgeR)
library(clusterProfiler)
library(org.Hs.eg.db)
library(ReactomePA)
library(reshape2)
library(gtools)
library(rrvgo)
library(tibble)

# data downloaded from https://hive.biochemistry.gwu.edu/biomuta on 2020 - 04
-
# 12
IGF2BP1_BioMuta <- read.csv(file =
"/Users/riccardomosca/Desktop/RAPseq_PAPER/FIGUREs/FIGURE7/NEW/BioMuta/1.biom
uta-proteinview-2020-04-12-15-35-25.csv",
    header = T, sep = ",")
IGF2BP1_BioMuta <- IGF2BP1_BioMuta[, c(1, 6, 7, 11)]
IGF2BP2_BioMuta <- read.csv(file =
"/Users/riccardomosca/Desktop/RAPseq_PAPER/FIGUREs/FIGURE7/NEW/BioMuta/2.biom
uta-proteinview-2020-04-12-15-50-03.csv",
    header = T, sep = ",")
IGF2BP2_BioMuta <- IGF2BP2_BioMuta[, c(1, 6, 7, 11)]
IGF2BP3_BioMuta <- read.csv(file =
"/Users/riccardomosca/Desktop/RAPseq_PAPER/FIGUREs/FIGURE7/NEW/BioMuta/3.biom
uta-proteinview-2020-04-12-15-51-10.csv",
    header = T, sep = ",")
IGF2BP3_BioMuta <- IGF2BP3_BioMuta[, c(1, 6, 7, 11)]

IGF2BP1 <-
read.table("/Users/riccardomosca/Desktop/RAPseq_PAPER/FIGUREs/FIGURE7/METAFIL
Es/SCALED/IGF2BP1_metafile_annotated.txt",
    sep = "\t", header = T)

IGF2BP2 <-
read.table("/Users/riccardomosca/Desktop/RAPseq_PAPER/FIGUREs/FIGURE7/METAFIL
```

```
Es/SCALED/IGF2BP2_metafile_annotated.txt",
    sep = "\t", header = T)
IGF2BP3 <-
read.table("/Users/riccardomosca/Desktop/RAPseq_PAPER/FIGUREs/FIGURE7/METAFIL
Es/SCALED/IGF2BP3_metafile_annotated.txt",
    sep = "\t", header = T)
```

Figure 5A

```
IGF2BPs_MUTs <- rbind(IGF2BP1_BioMuta,IGF2BP2_BioMuta,IGF2BP3_BioMuta)
IGF2BPs_MUTs$UniProtKB.AC <- gsub("Q9NZI8-
1","IGF2BP1",IGF2BPs_MUTs$UniProtKB.AC)
IGF2BPs_MUTs$UniProtKB.AC <- gsub("O00425-
1","IGF2BP3",IGF2BPs_MUTs$UniProtKB.AC)
IGF2BPs_MUTs$UniProtKB.AC <- gsub("Q9Y6M1-
2","IGF2BP2",IGF2BPs_MUTs$UniProtKB.AC)

IGF2BPs_MUTs <-
IGF2BPs_MUTs[,c("UniProtKB.AC","Protein.Position","Frequency")]
colnames(IGF2BPs_MUTs) <- c("gene_name","Protein.Position","Frequency")
IGF2BPs_MUTs <- tibble(IGF2BPs_MUTs)
IGF2BPs_MUTs <- IGF2BPs_MUTs %>% group_by(gene_name,Protein.Position) %>%
summarise(Freq = sum(Frequency))

## `summarise()` has grouped output by 'gene_name'. You can override using
the
## `.groups` argument.

Fig5A <- ggplot(data = IGF2BPs_MUTs) +
  geom_bar(aes(x=Protein.Position, y=Freq), stat="identity", size=0.5,
color="black") +
  geom_point(aes(x=Protein.Position, y=Freq, fill=gene_name), size=6, pch=21,
color="black") +
  facet_wrap(~gene_name) +
  scale_fill_manual(values = c("#E2C88A", "#E2C88A", "#E2C88A")) +
  theme_pubr() +
  ylim(0,40)

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

Fig5A
```

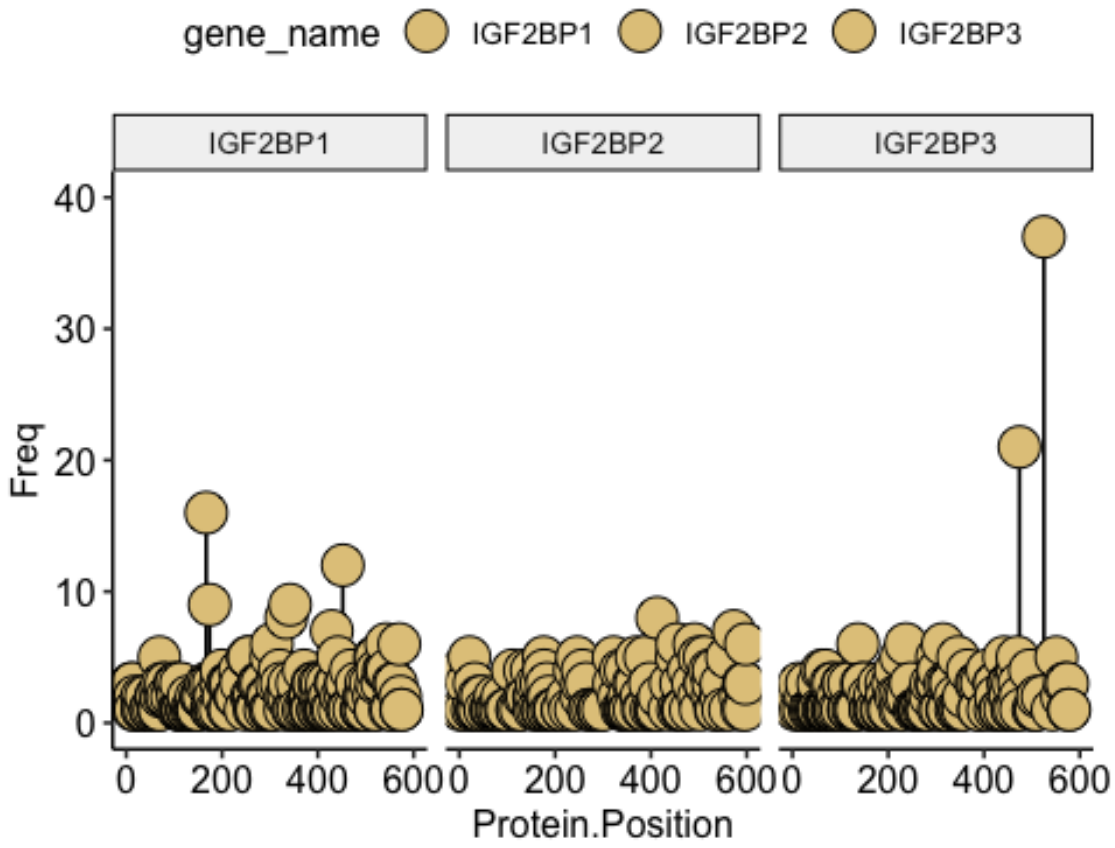
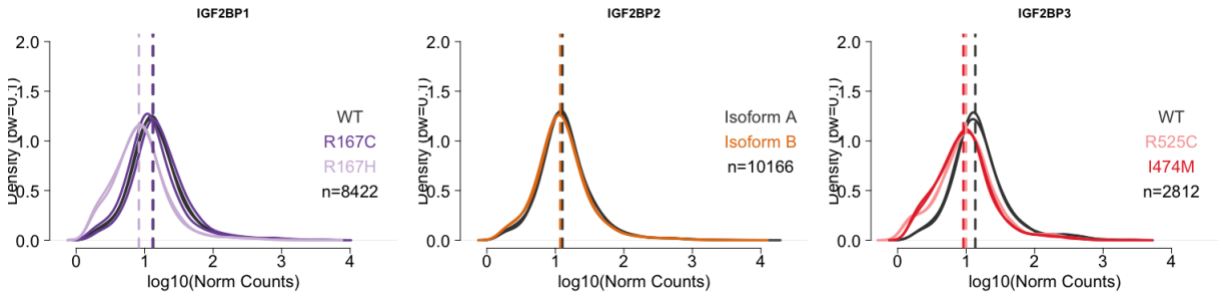


Figure 5C

```
IGF2BP1 <- IGF2BP1[IGF2BP1$IGF2BP1wt_rep1 >= 0.5 & IGF2BP1$IGF2BP1wt_rep2 >=
0.5 &
    IGF2BP1$IGF2BP1R167C_rep1 >= 0.5 & IGF2BP1$IGF2BP1R167C_rep2 >= 0.5 &
IGF2BP1$IGF2BP1R167H_rep1 >=
    0.5 & IGF2BP1$IGF2BP1R167H_rep2 >= 0.5, ]

par(mfrow = c(1, 3), bty = "n")
plot(density(log10(IGF2BP1$IGF2BP1wt_rep1 + 1), bw = 0.1), xlab = "log10(Norm
Counts)",
    ylab = "Density (bw=0.1)", col = "#404040", xlim = c(-0.2, 4.5), ylim =
c(0,
        2), lwd = 3, main = "IGF2BP1", las = 1, cex.axis = 2, cex.lab = 2,
cex.main = 1.5)
points(density(log10(IGF2BP1$IGF2BP1wt_rep2 + 1), bw = 0.1), type = "l", lwd
= 3,
    col = "#404040")
points(density(log10(IGF2BP1$IGF2BP1R167C_rep1 + 1), bw = 0.1), type = "l",
lwd = 3,
    col = "#7A52A5")
points(density(log10(IGF2BP1$IGF2BP1R167C_rep2 + 1), bw = 0.1), type = "l",
lwd = 3,
```

```
    col = "#7A52A5")
points(density(log10(IGF2BP1$IGF2BP1R167H_rep1 + 1), bw = 0.1), type = "l",
lwd = 3,
    col = "#D0BADA")
points(density(log10(IGF2BP1$IGF2BP1R167H_rep2 + 1), bw = 0.1), type = "l",
lwd = 3,
    col = "#D0BADA")
text(4, 1.25, "WT", col = "#404040", cex = 2)
text(4, 1, "R167C", col = "#7A52A5", cex = 2)
text(4, 0.75, "R167H", col = "#D0BADA", cex = 2)
text(4, 0.5, paste("n=", nrow(IGF2BP1), sep = ""), col = "black", cex = 2)
abline(v = median(log10(c(IGF2BP1$IGF2BP1wt_rep1 + 1, IGF2BP1$IGF2BP1wt_rep2
+ 1))),
    lwd = 3, col = "#404040", lty = 2)
abline(v = median(log10(c(IGF2BP1$IGF2BP1R167C_rep1 + 1,
IGF2BP1$IGF2BP1R167C_rep2 +
    1))), lwd = 3, col = "#7A52A5", lty = 2)
abline(v = median(log10(c(IGF2BP1$IGF2BP1R167H_rep1 + 1,
IGF2BP1$IGF2BP1R167H_rep2 +
    1))), lwd = 3, col = "#D0BADA", lty = 2)


IGF2BP2 <- IGF2BP2[IGF2BP2$IGF2BP2a_rep1 >= 0.5 & IGF2BP2$IGF2BP2a_rep2 >=
0.5 &
    IGF2BP2$IGF2BP2b_rep1 >= 0.5 & IGF2BP2$IGF2BP2b_rep2 >= 0.5, ]
plot(density(log10(IGF2BP2$IGF2BP2a_rep1 + 1), bw = 0.1), xlab = "log10(Norm
Counts)",
    ylab = "Density (bw=0.1)", col = "#404040", xlim = c(-0.2, 4.5), ylim =
c(0,
        2), lwd = 3, main = "IGF2BP2", las = 1, cex.axis = 2, cex.lab = 2,
cex.main = 1.5)
points(density(log10(IGF2BP2$IGF2BP2a_rep2 + 1), bw = 0.1), type = "l", lwd =
3,
    col = "#404040")
points(density(log10(IGF2BP2$IGF2BP2b_rep1 + 1), bw = 0.1), type = "l", lwd =
3,
    col = "#E47B12")
points(density(log10(IGF2BP2$IGF2BP2b_rep2 + 1), bw = 0.1), type = "l", lwd =
3,
    col = "#E47B12")
text(4, 1.25, "Isoform A", col = "#404040", cex = 2)
text(4, 1, "Isoform B", col = "#E47B12", cex = 2)
text(4, 0.75, paste("n=", nrow(IGF2BP2), sep = ""), col = "black", cex = 2)
abline(v = median(log10(c(IGF2BP2$IGF2BP2a_rep1 + 1, IGF2BP2$IGF2BP2a_rep2 +
1))),
    lwd = 3, col = "#404040", lty = 2)
abline(v = median(log10(c(IGF2BP2$IGF2BP2b_rep1 + 1, IGF2BP2$IGF2BP2b_rep2 +
1))),
    lwd = 3, col = "#E47B12", lty = 2)
```

```
IGF2BP3 <- IGF2BP3[IGF2BP3$IGF2BP3wt_rep1 >= 0.5 & IGF2BP3$IGF2BP3wt_rep2 >=
0.5 &
    IGF2BP3$IGF2BP3I474M_rep1 >= 0.5 & IGF2BP3$IGF2BP3I474M_rep2 >= 0.5 &
IGF2BP3$IGF2BP3R525C_rep1 &
    IGF2BP3$IGF2BP3R525C_rep2, ]
plot(density(log10(IGF2BP3$IGF2BP3wt_rep1 + 1), bw = 0.1), xlab = "log10(Norm
Counts)",
    ylab = "Density (bw=0.1)", col = "#404040", xlim = c(-0.2, 4.5), ylim =
c(0,
        2), lwd = 3, main = "IGF2BP3", las = 1, cex.axis = 2, cex.lab = 2,
cex.main = 1.5)
points(density(log10(IGF2BP3$IGF2BP3wt_rep2 + 1), bw = 0.1), type = "l", lwd
= 3,
    col = "#404040")
points(density(log10(IGF2BP3$IGF2BP3R525C_rep1 + 1), bw = 0.1), type = "l",
lwd = 3,
    col = "#FBA5A4")
points(density(log10(IGF2BP3$IGF2BP3R525C_rep2 + 1), bw = 0.1), type = "l",
lwd = 3,
    col = "#FBA5A4")
points(density(log10(IGF2BP3$IGF2BP3I474M_rep1 + 1), bw = 0.1), type = "l",
lwd = 3,
    col = "#E63234")
points(density(log10(IGF2BP3$IGF2BP3I474M_rep2 + 1), bw = 0.1), type = "l",
lwd = 3,
    col = "#E63234")
text(4, 1.25, "WT", col = "#404040", cex = 2)
text(4, 1, "R525C", col = "#FBA5A4", cex = 2)
text(4, 0.75, "I474M", col = "#E63234", cex = 2)
text(4, 0.5, paste("n=", nrow(IGF2BP3), sep = ""), col = "black", cex = 2)
abline(v = median(log10(c(IGF2BP3$IGF2BP3wt_rep1 + 1, IGF2BP3$IGF2BP3wt_rep2
+ 1))),
    lwd = 3, col = "#404040", lty = 2)
abline(v = median(log10(c(IGF2BP3$IGF2BP3R525C_rep1 + 1,
IGF2BP3$IGF2BP3R525C_rep2 +
    1))), lwd = 3, col = "#FBA5A4", lty = 2)
abline(v = median(log10(c(IGF2BP3$IGF2BP3I474M_rep1 + 1,
IGF2BP3$IGF2BP3I474M_rep2 +
    1))), lwd = 3, col = "#E63234", lty = 2)
```




```
t.test(c(IGF2BP1$IGF2BP1wt_rep2, IGF2BP1$IGF2BP1wt_rep1),
c(IGF2BP1$IGF2BP1R167C_rep1,
    IGF2BP1$IGF2BP1R167C_rep2))

##
##  Welch Two Sample t-test
##
## data:  c(IGF2BP1$IGF2BP1wt_rep2, IGF2BP1$IGF2BP1wt_rep1) and
c(IGF2BP1$IGF2BP1R167C_rep1, IGF2BP1$IGF2BP1R167C_rep2)
## t = -0.09462, df = 33656, p-value = 0.9246
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.293521  2.990178
## sample estimates:
## mean of x mean of y
##  33.10721  33.25888

t.test(c(IGF2BP1$IGF2BP1wt_rep2, IGF2BP1$IGF2BP1wt_rep1),
c(IGF2BP1$IGF2BP1R167H_rep1,
    IGF2BP1$IGF2BP1R167H_rep2))

##
##  Welch Two Sample t-test
##
## data:  c(IGF2BP1$IGF2BP1wt_rep2, IGF2BP1$IGF2BP1wt_rep1) and
c(IGF2BP1$IGF2BP1R167H_rep1, IGF2BP1$IGF2BP1R167H_rep2)
## t = 11.422, df = 28143, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  12.43142 17.58184
## sample estimates:
## mean of x mean of y
##  33.10721  18.10058

t.test(c(IGF2BP2$IGF2BP2a_rep1, IGF2BP2$IGF2BP2a_rep2),
c(IGF2BP2$IGF2BP2b_rep1,
    IGF2BP2$IGF2BP2b_rep2))

##
##  Welch Two Sample t-test
##
```

```
## data:  c(IGF2BP2$IGF2BP2a_rep1, IGF2BP2$IGF2BP2a_rep2) and
c(IGF2BP2$IGF2BP2b_rep1, IGF2BP2$IGF2BP2b_rep2)
## t = 2.4162, df = 40078, p-value = 0.01569
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.5824446 5.5875126
## sample estimates:
## mean of x mean of y
##  28.89451  25.80953

t.test(c(IGF2BP3$IGF2BP3wt_rep1, IGF2BP3$IGF2BP3wt_rep2),
c(IGF2BP3$IGF2BP3I474M_rep1,
    IGF2BP3$IGF2BP3I474M_rep2))

##
##  Welch Two Sample t-test
##
## data:  c(IGF2BP3$IGF2BP3wt_rep1, IGF2BP3$IGF2BP3wt_rep2) and
c(IGF2BP3$IGF2BP3I474M_rep1, IGF2BP3$IGF2BP3I474M_rep2)
## t = 8.382, df = 9695.8, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  10.53498 16.96642
## sample estimates:
## mean of x mean of y
##  32.06258  18.31187

t.test(c(IGF2BP3$IGF2BP3wt_rep1, IGF2BP3$IGF2BP3wt_rep2),
c(IGF2BP3$IGF2BP3R525C_rep1,
    IGF2BP3$IGF2BP3R525C_rep2))

##
##  Welch Two Sample t-test
##
## data:  c(IGF2BP3$IGF2BP3wt_rep1, IGF2BP3$IGF2BP3wt_rep2) and
c(IGF2BP3$IGF2BP3R525C_rep1, IGF2BP3$IGF2BP3R525C_rep2)
## t = 7.181, df = 9495.1, p-value = 7.442e-13
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   8.475634 14.840260
## sample estimates:
## mean of x mean of y
##  32.06258  20.40463
```

Figure 5D/S5F

```
IGF2BP1$Peak_ID <- paste(IGF2BP1$chr, IGF2BP1$start, IGF2BP1$end,
IGF2BP1$strand,
    sep = "_")
rownames(IGF2BP1) <- IGF2BP1$Peak_ID
SUB_IGF2BP1 <- IGF2BP1[, 6:11]
```

```
SUB_IGF2BP1 <- SUB_IGF2BP1[SUB_IGF2BP1$IGF2BP1wt_rep1 >= 5 &
SUB_IGF2BP1$IGF2BP1wt_rep2 >=
    5 & SUB_IGF2BP1$IGF2BP1R167C_rep1 >= 5 & SUB_IGF2BP1$IGF2BP1R167C_rep2 >=
5 &
    SUB_IGF2BP1$IGF2BP1R167H_rep1 >= 5 & SUB_IGF2BP1$IGF2BP1R167H_rep2 >= 5,
]

# Identification of the differentially bound sites between IGF2BP1 wt and
R167C
WTvsRtoC <- SUB_IGF2BP1[, c("IGF2BP1wt_rep1", "IGF2BP1wt_rep2",
"IGF2BP1R167C_rep1",
    "IGF2BP1R167C_rep2")]
x <- WTvsRtoC
group <- c(1, 1, 2, 2)
y <- DGEList(counts = x, group = group)
design <- model.matrix(~group)
y <- estimateDisp(y, design)
y$samples$lib.size <- c(1e+06, 1e+06, 1e+06, 1e+06)
fit <- glmQLFit(y, design)
qlf <- glmQLFTest(fit)
X <- qlf$table
X$p.adjust <- -log10(p.adjust(X$PValue, method = "BH"))
X$PValue <- -log10(X$PValue)
WTvsRtoC$FDR <- X$p.adjust
WTvsRtoC$Significant <- WTvsRtoC$FDR >= 1.30103
WTvsRtoC$log2FC <- X$logFC
WTvsRtoC$log2CPM <- X$logCPM
WTvsRtoC$wt_won <- WTvsRtoC$log2FC <= -1 & WTvsRtoC$FDR >= 1.30103
WTvsRtoC$RtoC_won <- WTvsRtoC$log2FC >= 1 & WTvsRtoC$FDR >= 1.30103
WTvsRtoC$DiffBind <- paste(WTvsRtoC$wt_won, WTvsRtoC$RtoC_won, sep = "_")
WTvsRtoC$DiffBind <- gsub("FALSE_FALSE", "N.S.", WTvsRtoC$DiffBind)
WTvsRtoC$DiffBind <- gsub("FALSE_TRUE", "R167C", WTvsRtoC$DiffBind)
WTvsRtoC$DiffBind <- gsub("TRUE_FALSE", "wt", WTvsRtoC$DiffBind)



# Identification of the differentially bound sites between IGF2BP1 wt and
R167H
WTvsRtoH <- SUB_IGF2BP1[, c("IGF2BP1wt_rep1", "IGF2BP1wt_rep2",
"IGF2BP1R167H_rep1",
    "IGF2BP1R167H_rep2")]
colnames(WTvsRtoH) <- c("IGF2BP1wt_rep1", "IGF2BP1wt_rep2",
"IGF2BP1R167H_rep1",
    "IGF2BP1R167H_rep2")
x <- WTvsRtoH
group <- c(1, 1, 2, 2)
y <- DGEList(counts = x, group = group)
design <- model.matrix(~group)
y <- estimateDisp(y, design)
```

```
y$samples$lib.size <- c(1e+06, 1e+06, 1e+06, 1e+06)
fit <- glmQLFit(y, design)
qlf <- glmQLFTest(fit)
X <- qlf$table
X$p.adjust <- -log10(p.adjust(X$PValue, method = "BH"))
X$PValue <- -log10(X$PValue)
WTvsRtoH$FDR <- X$p.adjust
WTvsRtoH$Significant <- WTvsRtoH$FDR >= 1.30103
WTvsRtoH$log2FC <- X$logFC
WTvsRtoH$log2CPM <- X$logCPM
WTvsRtoH$wt_won <- WTvsRtoH$log2FC <= -1 & WTvsRtoH$FDR >= 1.30103
WTvsRtoH$RtoH_won <- WTvsRtoH$log2FC >= 1 & WTvsRtoH$FDR >= 1.30103
WTvsRtoH$DiffBind <- paste(WTvsRtoH$wt_won, WTvsRtoH$RtoH_won, sep = "_")
WTvsRtoH$DiffBind <- gsub("FALSE_FALSE", "N.S.", WTvsRtoH$DiffBind)
WTvsRtoH$DiffBind <- gsub("FALSE_TRUE", "R167H", WTvsRtoH$DiffBind)
WTvsRtoH$DiffBind <- gsub("TRUE_FALSE", "wt", WTvsRtoH$DiffBind)
WTvsRtoC <- WTvsRtoC[, c(5, 7, 8, 11)]
WTvsRtoC$comparison <- rep("wtBP1 vs R167C", nrow(WTvsRtoC))
WTvsRtoH <- WTvsRtoH[, c(5, 7, 8, 11)]
WTvsRtoH$comparison <- rep("wtBP1 vs R167H", nrow(WTvsRtoH))
COMPARISONS_1 <- rbind(WTvsRtoC, WTvsRtoH)
colnames(COMPARISONS_1) <- c("minuslog10FDR", "log2FC", "log2CPM",
"DiffBind", "comparison")

WTvsRtoC <- rownames_to_column(WTvsRtoC, var = "Peak_ID")
WTvsRtoC <- merge(WTvsRtoC, IGF2BP1, by = "Peak_ID")



WTvsRtoH <- rownames_to_column(WTvsRtoH, var = "Peak_ID")
WTvsRtoH <- merge(WTvsRtoH, IGF2BP1, by = "Peak_ID")



IGF2BP2$Peak_ID <- paste(IGF2BP2$chr, IGF2BP2$start, IGF2BP2$end,
IGF2BP2$strand,
    sep = "_")
rownames(IGF2BP2) <- IGF2BP2$Peak_ID
SUB_IGF2BP2 <- IGF2BP2[, 6:9, drop = FALSE]
SUB_IGF2BP2 <- SUB_IGF2BP2[SUB_IGF2BP2$IGF2BP2a_rep1 >= 5 &
SUB_IGF2BP2$IGF2BP2a_rep2 >=
    5 & SUB_IGF2BP2$IGF2BP2b_rep1 >= 5 & SUB_IGF2BP2$IGF2BP2b_rep2 >= 5, ]

# Identification of the differentially bound sites between IGF2BP2 isoform A
# and B
AvsB <- SUB_IGF2BP2
x <- AvsB
group <- c(1, 1, 2, 2)
y <- DGEList(counts = x, group = group)
design <- model.matrix(~group)
y <- estimateDisp(y, design)
```

```r
y$samples$lib.size <- c(1e+06, 1e+06, 1e+06, 1e+06)
fit <- glmQLFit(y, design)
qlf <- glmQLFTest(fit)
X <- qlf$table
X$p.adjust <- -log10(p.adjust(X$PValue, method = "BH"))
X$PValue <- -log10(X$PValue)
AvsB$FDR <- X$p.adjust
AvsB$Significant <- AvsB$FDR >= 1.30103
AvsB$log2FC <- X$logFC
AvsB$log2CPM <- X$logCPM
AvsB$A_won <- AvsB$log2FC <= -1 & AvsB$FDR >= 1.30103
AvsB$B_won <- AvsB$log2FC >= 1 & AvsB$FDR >= 1.30103
AvsB$DiffBind <- paste(AvsB$A_won, AvsB$B_won, sep = "_")
AvsB$DiffBind <- gsub("FALSE_FALSE", "N.S.", AvsB$DiffBind)
AvsB$DiffBind <- gsub("FALSE_TRUE", "Isoform_B", AvsB$DiffBind)
AvsB$DiffBind <- gsub("TRUE_FALSE", "Isoform_A", AvsB$DiffBind)
AvsB$comparison <- rep("Isof_A vs Isof_B", nrow(AvsB))
COMPARISONS_2 <- AvsB[, c(5, 7, 8, 11, 12)]
colnames(COMPARISONS_2) <- c("minuslog10FDR", "log2FC", "log2CPM",
"DiffBind", "comparison")

AvsB <- rownames_to_column(AvsB, var = "Peak_ID")
AvsB <- merge(AvsB, IGF2BP2, by = "Peak_ID")



IGF2BP3$Peak_ID <- paste(IGF2BP3$chr, IGF2BP3$start, IGF2BP3$end,
IGF2BP3$strand,
    sep = "_")
rownames(IGF2BP3) <- IGF2BP3$Peak_ID
SUB_IGF2BP3 <- IGF2BP3[, 6:11]
SUB_IGF2BP3 <- SUB_IGF2BP3[SUB_IGF2BP3$IGF2BP3wt_rep1 >= 5 &
SUB_IGF2BP3$IGF2BP3wt_rep2 >=
    5 & SUB_IGF2BP3$IGF2BP3R525C_rep1 >= 5 & SUB_IGF2BP3$IGF2BP3R525C_rep2 >=
5 &
    SUB_IGF2BP3$IGF2BP3I474M_rep1 >= 5 & SUB_IGF2BP3$IGF2BP3I474M_rep2 >= 5,
]

# Identification of the differentially bound sites between IGF2BP3 wt and
R525C
WTvsR525C <- SUB_IGF2BP3[, c("IGF2BP3wt_rep1", "IGF2BP3wt_rep1",
"IGF2BP3R525C_rep1",
    "IGF2BP3R525C_rep2")]
x <- WTvsR525C
group <- c(1, 1, 2, 2)
y <- DGEList(counts = x, group = group)
design <- model.matrix(~group)
y <- estimateDisp(y, design)
y$samples$lib.size <- c(1e+06, 1e+06, 1e+06, 1e+06)
fit <- glmQLFit(y, design)
```

```r
qlf <- glmQLFTest(fit)
X <- qlf$table
X$p.adjust <- -log10(p.adjust(X$PValue, method = "BH"))
X$PValue <- -log10(X$PValue)
WTvsR525C$FDR <- X$p.adjust
WTvsR525C$Significant <- WTvsR525C$FDR >= 1.30103
WTvsR525C$log2FC <- X$logFC
WTvsR525C$log2CPM <- X$logCPM
WTvsR525C$wt_won <- WTvsR525C$log2FC <= -1 & WTvsR525C$FDR >= 1.30103
WTvsR525C$R525C_won <- WTvsR525C$log2FC >= 1 & WTvsR525C$FDR >= 1.30103
WTvsR525C$DiffBind <- paste(WTvsR525C$wt_won, WTvsR525C$R525C_won, sep = "_")
WTvsR525C$DiffBind <- gsub("FALSE_FALSE", "N.S.", WTvsR525C$DiffBind)
WTvsR525C$DiffBind <- gsub("FALSE_TRUE", "R525C", WTvsR525C$DiffBind)
WTvsR525C$DiffBind <- gsub("TRUE_FALSE", "wt", WTvsR525C$DiffBind)




# Identification of the differentially bound sites between IGF2BP3 wt and
I474M
WTvsI474M <- SUB_IGF2BP3[, c("IGF2BP3wt_rep1", "IGF2BP3wt_rep2",
"IGF2BP3I474M_rep1",
    "IGF2BP3I474M_rep2")]
x <- WTvsI474M
group <- c(1, 1, 2, 2)
y <- DGEList(counts = x, group = group)
design <- model.matrix(~group)
y <- estimateDisp(y, design)
y$samples$lib.size <- c(1e+06, 1e+06, 1e+06, 1e+06)
fit <- glmQLFit(y, design)
qlf <- glmQLFTest(fit)
X <- qlf$table
X$p.adjust <- -log10(p.adjust(X$PValue, method = "BH"))
X$PValue <- -log10(X$PValue)
WTvsI474M$FDR <- X$p.adjust
WTvsI474M$Significant <- WTvsI474M$FDR >= 1.30103
WTvsI474M$log2FC <- X$logFC
WTvsI474M$log2CPM <- X$logCPM
WTvsI474M$wt_won <- WTvsI474M$log2FC <= -1 & WTvsI474M$FDR >= 1.30103
WTvsI474M$I474M_won <- WTvsI474M$log2FC >= 1 & WTvsI474M$FDR >= 1.30103
WTvsI474M$DiffBind <- paste(WTvsI474M$wt_won, WTvsI474M$I474M_won, sep = "_")
WTvsI474M$DiffBind <- gsub("FALSE_FALSE", "N.S.", WTvsI474M$DiffBind)
WTvsI474M$DiffBind <- gsub("FALSE_TRUE", "I474M", WTvsI474M$DiffBind)
WTvsI474M$DiffBind <- gsub("TRUE_FALSE", "wt", WTvsI474M$DiffBind)
WTvsR525C <- WTvsR525C[, c(5, 7, 8, 11)]
WTvsR525C$comparison <- rep("wtBP3 vs R525C", nrow(WTvsR525C))
WTvsI474M <- WTvsI474M[, c(5, 7, 8, 11)]
WTvsI474M$comparison <- rep("wtBP3 vs I474M", nrow(WTvsI474M))
COMPARISONS_3 <- rbind(WTvsR525C, WTvsI474M)
```

```r
colnames(COMPARISONS_3) <- c("minuslog10FDR", "log2FC", "log2CPM",
"DiffBind", "comparison")

WTvsR525C <- rownames_to_column(WTvsR525C, var = "Peak_ID")
WTvsR525C <- merge(WTvsR525C, IGF2BP3, by = "Peak_ID")

WTvsI474M <- rownames_to_column(WTvsI474M, var = "Peak_ID")
WTvsI474M <- merge(WTvsI474M, IGF2BP3, by = "Peak_ID")



ALL_COMPARISONS <- rbind(COMPARISONS_1, COMPARISONS_2, COMPARISONS_3)
ALL_COMPARISONS$comparison <- factor(ALL_COMPARISONS$comparison, levels =
c("wtBP1 vs R167C",
    "wtBP1 vs R167H", "Isof_A vs Isof_B", "wtBP3 vs I474M", "wtBP3 vs
R525C"))

ALL_COMPARISONS$DiffBind[ALL_COMPARISONS$DiffBind == "Isoform_A"] <- "wt"
table(ALL_COMPARISONS[, c("DiffBind", "comparison")])
```

```
##            comparison
## DiffBind    wtBP1 vs R167C wtBP1 vs R167H Isof_A vs Isof_B wtBP3 vs I474M
##   I474M                 0              0               0              3
##   Isoform_B             0              0              41              0
##   N.S.               3203           2697            5752            817
##   R167C                 1              0               0              0
##   R167H                 0             15               0              0
##   R525C                 0              0               0              0
##   wt                    1            493              68             97
##            comparison
## DiffBind    wtBP3 vs R525C
##   I474M                 0
##   Isoform_B             0
##   N.S.                806
##   R167C                 0
##   R167H                 0
##   R525C                12
##   wt                   99
```

```r
Fig5D <- ggplot(data = ALL_COMPARISONS) + geom_point(aes(x = log2CPM, y =
log2FC,
    color = DiffBind), size = 3) + scale_color_manual(values = c("#E31A1C",
"#E47B12",
    "#808080", "#6A3D9A", "#CAB2D6", "#FB9A99", "#404040")) + theme_pubr() +
geom_hline(yintercept = 0,
    size = 1, color = "black", lty = 2) + ylim(-8, 8) + xlim(2, 15) +
facet_wrap(~comparison,
    ncol = 5)

Fig5D
```

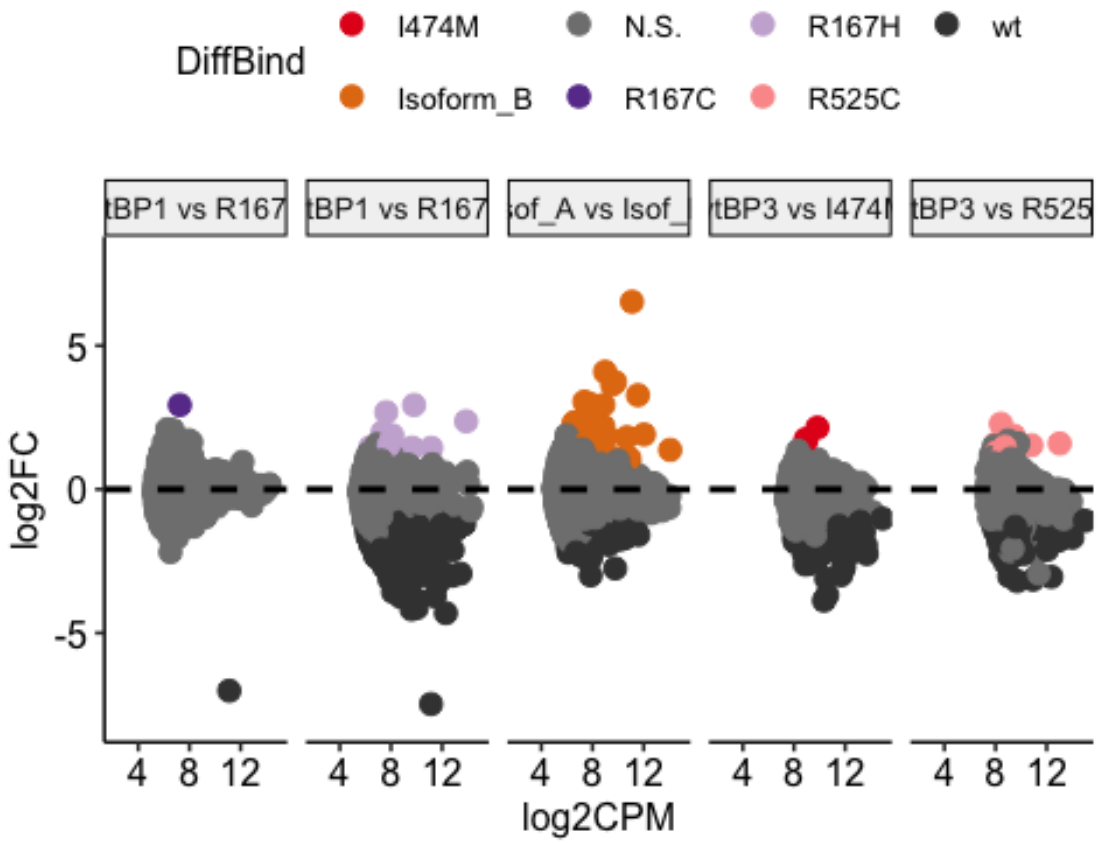
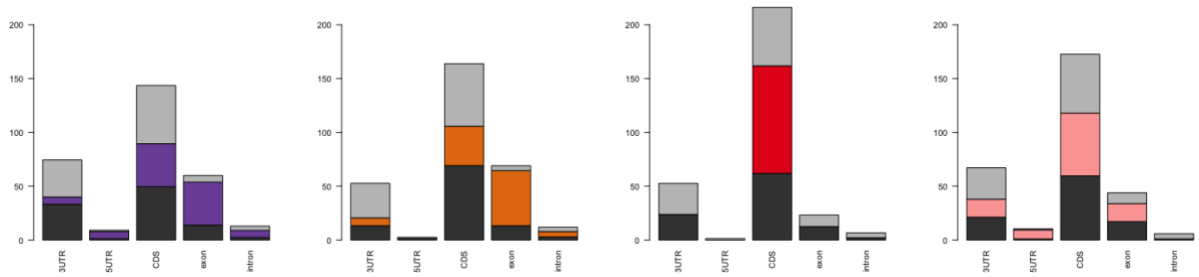


Figure 5E/S5G

```
par(mfrow = c(1, 4))


WTvsRtoH$DiffBind <- factor(WTvsRtoH$DiffBind, levels = c("wt", "R167H",
"N.S."))
barplot((t(table(WTvsRtoH[, c("feature",
"DiffBind")]))/rowSums(t(table(WTvsRtoH[,
    c("feature", "DiffBind")])))) * 100, las = 2, col = c("#404040",
"#7B52A6", "#C0C0C0"),
    ylim = c(0, 200))




AvsB$DiffBind <- factor(AvsB$DiffBind, levels = c("Isoform_A", "Isoform_B",
"N.S."))
barplot((t(table(AvsB[, c("feature", "DiffBind")]))/rowSums(t(table(AvsB[,
c("feature",
    "DiffBind")])))) * 100, las = 2, col = c("#404040", "#E47B12",
"#C0C0C0"), ylim = c(0,
```

```
    200))



WTvsI474M$DiffBind <- factor(WTvsI474M$DiffBind, levels = c("wt", "I474M",
"N.S."))
barplot((t(table(WTvsI474M[, c("feature",
"DiffBind")]))/rowSums(t(table(WTvsI474M[,
    c("feature", "DiffBind")])))) * 100, las = 2, col = c("#404040",
"#E31A1C", "#C0C0C0"),
    ylim = c(0, 200))



WTvsR525C$DiffBind <- factor(WTvsR525C$DiffBind, levels = c("wt", "R525C",
"N.S."))
barplot((t(table(WTvsR525C[, c("feature",
"DiffBind")]))/rowSums(t(table(WTvsR525C[,
    c("feature", "DiffBind")])))) * 100, las = 2, col = c("#404040",
"#FBA5A4", "#C0C0C0"),
    ylim = c(0, 200))
```




```
# Statistics IGF2BP1

# 3UTR vs CDS 0.0056
options(scipen = 99)
aa <- (t(table(WTvsRtoH[, c("feature",
"DiffBind")]))/rowSums(t(table(WTvsRtoH[,
    c("feature", "DiffBind")])))) * 100
aa <- aa[2:3, c(1, 3)]
ThreeUTRvsCDS_IGF2BP1 <- chisq.test(aa)

# 3UTR vs exon 10^-7
options(scipen = 99)
aa <- (t(table(WTvsRtoH[, c("feature",
"DiffBind")]))/rowSums(t(table(WTvsRtoH[,
    c("feature", "DiffBind")])))) * 100
aa <- aa[2:3, c(1, 4)]
```

```r
ThreeUTRvsexon_IGF2BP1 <- chisq.test(aa)

# CDS vs exon 0.000002
options(scipen = 99)
aa <- (t(table(WTvsRtoH[, c("feature",
"DiffBind")]))/rowSums(t(table(WTvsRtoH[,
    c("feature", "DiffBind")])))) * 100
aa <- aa[2:3, c(3, 4)]
CDSvsexon_IGF2BP1 <- chisq.test(aa)


# Statistics IGFBP2

# CDS vs exon 10^-8
options(scipen = 99)
aa <- (t(table(AvsB[, c("feature", "DiffBind")]))/rowSums(t(table(AvsB[,
c("feature",
    "DiffBind")])))) * 100
aa <- aa[1:2, c(3, 4)]
CDSvsexon_IGF2BP2 <- chisq.test(aa)

# 3UTR vs ncRNA 0.0005
aa <- (t(table(AvsB[, c("feature", "DiffBind")]))/rowSums(t(table(AvsB[,
c("feature",
    "DiffBind")])))) * 100
aa <- aa[1:2, c(1, 4)]
ThreeUTRvsexon_IGF2BP2 <- chisq.test(aa)


# Statistics IGF2BP3 I474M

# 3UTR vs CDS 2 *10^-8
options(scipen = 99)
aa <- (t(table(WTvsI474M[, c("feature",
"DiffBind")]))/rowSums(t(table(WTvsI474M[,
    c("feature", "DiffBind")])))) * 100
aa <- aa[1:3, c(1, 3)]
ThreeUTRvsCDS_IGF2BP3_I474M <- chisq.test(aa)

# CDS vs exon 0.0001
options(scipen = 99)
aa <- (t(table(WTvsI474M[, c("feature",
"DiffBind")]))/rowSums(t(table(WTvsI474M[,
    c("feature", "DiffBind")])))) * 100
aa <- aa[1:3, c(3, 4)]
CSDvsExon_IGF2BP3_I474M <- chisq.test(aa)

# Statistics IGF2BP3 R525C
```

```r
# 3UTR vs CDS ns
options(scipen = 99)
aa <- (t(table(WTvsR525C[, c("feature",
"DiffBind")]))/rowSums(t(table(WTvsR525C[,
    c("feature", "DiffBind")])))) * 100
aa <- aa[2:3, c(1, 3)]
ThreeUTRvsCDS_IGF2BP3_R525C <- chisq.test(aa)

# CDS vs exon ns
options(scipen = 99)
aa <- (t(table(WTvsR525C[, c("feature",
"DiffBind")]))/rowSums(t(table(WTvsR525C[,
    c("feature", "DiffBind")])))) * 100
aa <- aa[1:3, c(3, 4)]
CDSvsexon_IGF2BP3_R525C <- chisq.test(aa)
```

Figure 5F

```r
SUB <- AvsB[AvsB$DiffBind != "N.S.", ]
SUB$DiffBind <- factor(SUB$DiffBind, levels = c("Isoform_A", "Isoform_B"))
SUB <- SUB[SUB$gene_type != "protein_coding", ]  # ncRNAs comparison
ncRNA_enrichment <- t(table(SUB[, c("gene_type", "DiffBind")]))


barplot(ncRNA_enrichment[, order(-ncRNA_enrichment[2, ])], las = 2, col =
c("#404040",
    "#E47B12"))
```

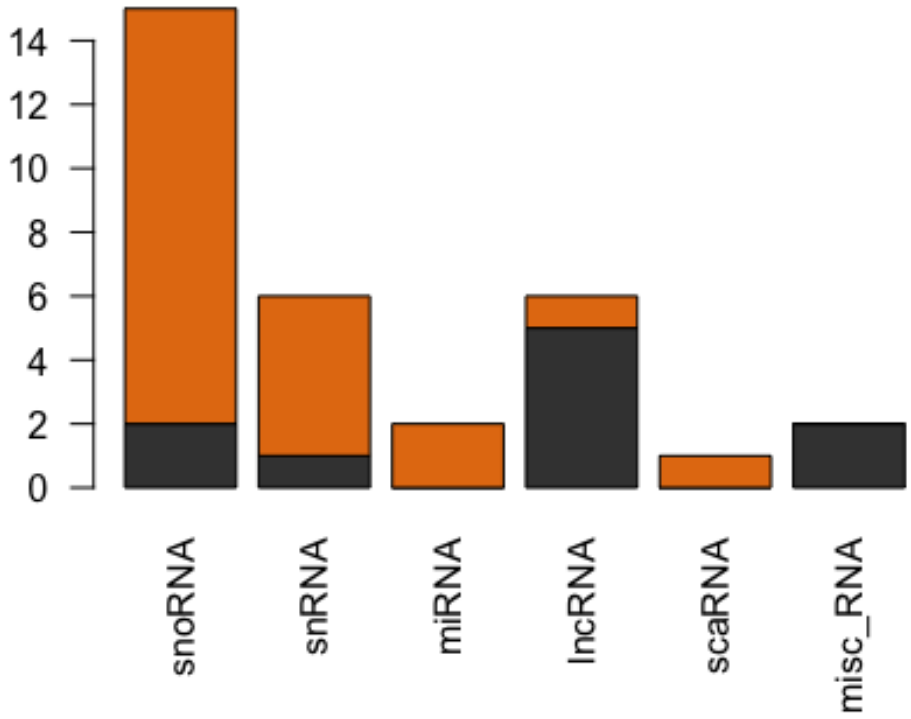


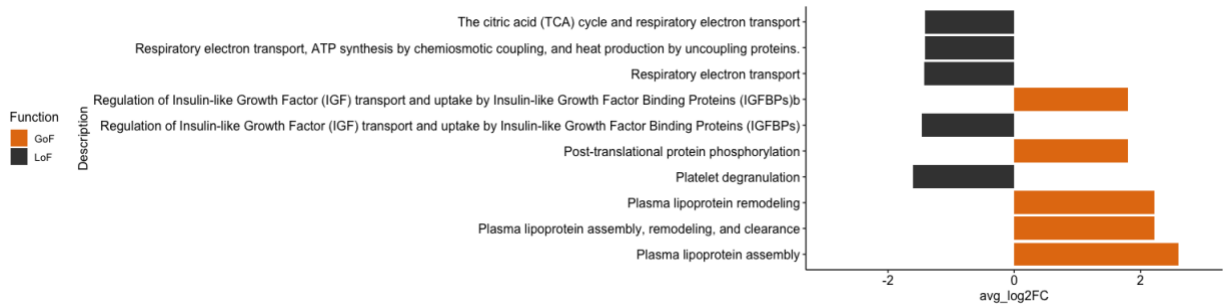
Figure 5H

```
IGF2BP2a_diff_bound <- AvsB[AvsB$DiffBind == "Isoform_A", ]
IGF2BP2b_diff_bound <- AvsB[AvsB$DiffBind == "Isoform_B", ]



IGF2BP2a_diff_bound <- IGF2BP2a_diff_bound %>%
    mutate(gene_ID = strsplit(gene_ID, "\\.") %>%
        lapply(., function(x) x[1]) %>%
        unlist())

IGF2BP2b_diff_bound <- IGF2BP2b_diff_bound %>%
    mutate(gene_ID = strsplit(gene_ID, "\\.") %>%
        lapply(., function(x) x[1]) %>%
        unlist())

IGF2BP2a_diff_bound_gene_ID <- subset(IGF2BP2a_diff_bound, select = c(23))
IGF2BP2b_diff_bound_gene_ID <- subset(IGF2BP2b_diff_bound, select = c(23))

Inputs <- read.table(file =
"/Users/riccardomosca/Desktop/RAPseq_PAPER/Inputs_TPM.txt",
```

```
    stringsAsFactors = F, header = T)

IDs_isofA <- mapIds(x = org.Hs.eg.db, keys =
IGF2BP2a_diff_bound_gene_ID$gene_ID,
    column = "ENTREZID", keytype = "ENSEMBL")
IDs_isofA <- as.character(IDs_isofA)[is.na(as.character(IDs_isofA)) == FALSE]
IDs_background <- mapIds(x = org.Hs.eg.db, keys = Inputs$Geneid, column =
"ENTREZID",
    keytype = "ENSEMBL")
IDs_background <-
as.character(IDs_background)[is.na(as.character(IDs_background)) ==
    FALSE]

# Pathway analysis IGF2BP2 isoform A
Reactome_a <- enrichPathway(gene = IDs_isofA, organism = "human",
pAdjustMethod = "BH",
    universe = IDs_background, readable = FALSE)

IDs_isofB <- mapIds(x = org.Hs.eg.db, keys =
IGF2BP2b_diff_bound_gene_ID$gene_ID,
    column = "ENTREZID", keytype = "ENSEMBL")
IDs_isofB <- as.character(IDs_isofB)[is.na(as.character(IDs_isofB)) == FALSE]

# Pathway analysis IGF2BP2 isoform B
Reactome_b <- enrichPathway(gene = IDs_isofB, organism = "human",
pAdjustMethod = "BH",
    universe = IDs_background, readable = FALSE)

Pathways_a <- as.data.frame(Reactome_a)
Pathways_b <- as.data.frame(Reactome_b)

top_5_a <- Pathways_a[1:5, ]

top_5_a <- top_5_a %>%
    mutate(gene_list = strsplit(as.character(geneID), "/"))

avg_fc_list <- list()


for (i in seq_len(nrow(top_5_a))) {
    genes <- unlist(top_5_a$gene_list[i])
    IDs <- mapIds(x = org.Hs.eg.db, keys = genes, column = "ENSEMBL", keytype
= "ENTREZID")
    # Step 4: Filter IGF2BP2a_diff_bound for matching genes and get log2FC
    matching_fc <- IGF2BP2a_diff_bound %>%
        filter(gene_ID %in% IDs) %>%
        pull(log2FC)  # Extract fold change values
```

```r
    # Compute average log2FC
    avg_fc <- ifelse(length(matching_fc) > 0, mean(matching_fc, na.rm =
TRUE), NA)

    avg_fc_list[[i]] <- avg_fc
}


top_5_a$avg_log2FC <- unlist(avg_fc_list)




top_5_b <- Pathways_b[1:5, ]

top_5_b <- top_5_b %>%
    mutate(gene_list = strsplit(as.character(geneID), "/"))

avg_fc_list <- list()


for (i in seq_len(nrow(top_5_b))) {
    genes <- unlist(top_5_b$gene_list[i])
    IDs <- mapIds(x = org.Hs.eg.db, keys = genes, column = "ENSEMBL", keytype
= "ENTREZID")
    # Step 4: Filter second dataframe for matching genes and get log2FC
    matching_fc <- IGF2BP2b_diff_bound %>%
        filter(gene_ID %in% IDs) %>%
        pull(log2FC)  # Extract fold change values

    # Compute average log2FC
    avg_fc <- ifelse(length(matching_fc) > 0, mean(matching_fc, na.rm =
TRUE), NA)

    # Store result
    avg_fc_list[[i]] <- avg_fc
}


top_5_b$avg_log2FC <- unlist(avg_fc_list)




top10_a_b <- top_5_a %>%
    full_join(top_5_b) %>%
```

```
    mutate(Function = ifelse(avg_log2FC > 0, "GoF", "LoF"))

top10_a_b$Description[7] <- paste0(top10_a_b$Description[7], "b")

Fig5H <- ggplot(data = top10_a_b, aes(x = Description, y = avg_log2FC, fill =
Function)) +
    geom_bar(stat = "identity") + coord_flip() + scale_alpha_binned(range =
c(0.5,
    1)) + scale_fill_manual(values = c("#E47B12", "#404040")) + ylim(-3, 3) +
theme_pubr(legend = "left")

Fig5H
```

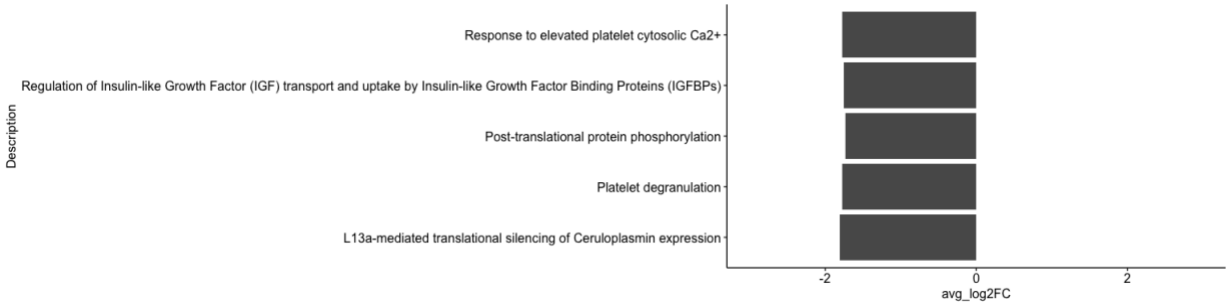


Check pathway enrichment in the mutants

```
IGF2BP1wt_diff_bound <- WTvsRtoH[WTvsRtoH$DiffBind == "wt", ]
IGF2BP1R167H_diff_bound <- WTvsRtoH[WTvsRtoH$DiffBind == "R167H", ]



IGF2BP1wt_diff_bound <- IGF2BP1wt_diff_bound %>%
    mutate(gene_ID = strsplit(gene_ID, "\\.") %>%
        lapply(., function(x) x[1]) %>%
        unlist())

IGF2BP1R167H_diff_bound <- IGF2BP1R167H_diff_bound %>%
    mutate(gene_ID = strsplit(gene_ID, "\\.") %>%
        lapply(., function(x) x[1]) %>%
        unlist())

IGF2BP1wt_diff_bound_gene_ID <- subset(IGF2BP1wt_diff_bound, select = c(18))
IGF2BP1R167H_diff_bound_gene_ID <- subset(IGF2BP1R167H_diff_bound, select =
c(18))

Inputs <- read.table(file =
"/Users/riccardomosca/Desktop/RAPseq_PAPER/Inputs_TPM.txt",
    stringsAsFactors = F, header = T)

IDs_wt <- mapIds(x = org.Hs.eg.db, keys =
IGF2BP1wt_diff_bound_gene_ID$gene_ID, column = "ENTREZID",
```

```
    keytype = "ENSEMBL")
IDs_wt <- as.character(IDs_wt)[is.na(as.character(IDs_wt)) == FALSE]
IDs_background <- mapIds(x = org.Hs.eg.db, keys = Inputs$Geneid, column =
"ENTREZID",
    keytype = "ENSEMBL")
IDs_background <-
as.character(IDs_background)[is.na(as.character(IDs_background)) ==
    FALSE]

# Pathway analysis IGF2BP1 wt
Reactome_wt_IGF1 <- enrichPathway(gene = IDs_wt, organism = "human",
pAdjustMethod = "BH",
    universe = IDs_background, readable = FALSE)

IDs_R167H <- mapIds(x = org.Hs.eg.db, keys =
IGF2BP1R167H_diff_bound_gene_ID$gene_ID,
    column = "ENTREZID", keytype = "ENSEMBL")
IDs_R167H <- as.character(IDs_R167H)[is.na(as.character(IDs_R167H)) == FALSE]

# Pathway analysis IGF2BP1R167H
Reactome_R167H <- enrichPathway(gene = IDs_R167H, organism = "human",
pAdjustMethod = "BH",
    universe = IDs_background, readable = FALSE)

Pathways_wt_IGF1 <- as.data.frame(Reactome_wt_IGF1)



top_5_wt_IGF1 <- Pathways_wt_IGF1[1:5, ]

top_5_wt_IGF1 <- top_5_wt_IGF1 %>%
    mutate(gene_list = strsplit(as.character(geneID), "/"))

avg_fc_list <- list()


for (i in seq_len(nrow(top_5_wt_IGF1))) {
    genes <- unlist(top_5_wt_IGF1$gene_list[i])
    IDs <- mapIds(x = org.Hs.eg.db, keys = genes, column = "ENSEMBL", keytype
= "ENTREZID")
    # Step 4: Filter IGF2BP2a_diff_bound for matching genes and get log2FC
    matching_fc <- IGF2BP1wt_diff_bound %>%
        filter(gene_ID %in% IDs) %>%
        pull(log2FC)  # Extract fold change values

    # Compute average log2FC
    avg_fc <- ifelse(length(matching_fc) > 0, mean(matching_fc, na.rm =
TRUE), NA)
```

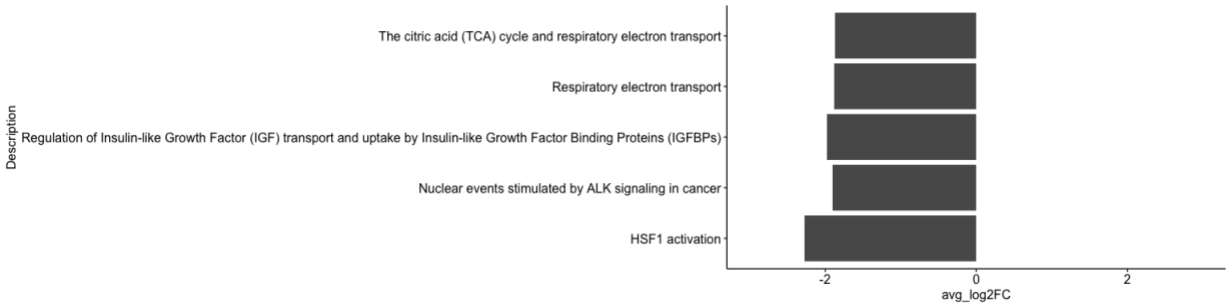
```
    avg_fc_list[[i]] <- avg_fc
}


top_5_wt_IGF1$avg_log2FC <- unlist(avg_fc_list)

FigIGF1_wt <- ggplot(data = top_5_wt_IGF1, aes(x = Description, y =
avg_log2FC)) +
    geom_bar(stat = "identity") + coord_flip() + scale_alpha_binned(range =
c(0.5,
    1)) + ylim(-3, 3) + theme_pubr(legend = "left")

FigIGF1_wt
```




```
# IGF2BP3 WTvsI474M
IGF2BP3wt_diff_bound <- WTvsI474M[WTvsI474M$DiffBind == "wt", ]
IGF2BP3WTvsI474M_diff_bound <- WTvsI474M[WTvsI474M$DiffBind == "I474M", ]



IGF2BP3wt_diff_bound <- IGF2BP3wt_diff_bound %>%
    mutate(gene_ID = strsplit(gene_ID, "\\.") %>%
        lapply(., function(x) x[1]) %>%
        unlist())

IGF2BP3WTvsI474M_diff_bound <- IGF2BP3WTvsI474M_diff_bound %>%
    mutate(gene_ID = strsplit(gene_ID, "\\.") %>%
        lapply(., function(x) x[1]) %>%
        unlist())

IGF2BP3wt_diff_bound_gene_ID <- subset(IGF2BP3wt_diff_bound, select = c(18))
IGF2BP3WTvsI474M_diff_bound_gene_ID <- subset(IGF2BP3WTvsI474M_diff_bound,
select = c(18))



IDs_wt <- mapIds(x = org.Hs.eg.db, keys =
IGF2BP3wt_diff_bound_gene_ID$gene_ID, column = "ENTREZID",
    keytype = "ENSEMBL")
```

```r
IDs_wt <- as.character(IDs_wt)[is.na(as.character(IDs_wt)) == FALSE]
IDs_background <- mapIds(x = org.Hs.eg.db, keys = Inputs$Geneid, column =
"ENTREZID",
    keytype = "ENSEMBL")
IDs_background <-
as.character(IDs_background)[is.na(as.character(IDs_background)) ==
    FALSE]

# Pathway analysis IGF2BP3 wt
Reactome_wt_IGF3_I474M <- enrichPathway(gene = IDs_wt, organism = "human",
pAdjustMethod = "BH",
    universe = IDs_background, readable = FALSE)

Pathways_wt_IGF3_I474M <- as.data.frame(Reactome_wt_IGF3_I474M)



top_5_wt_IGF3_I474M <- Pathways_wt_IGF3_I474M[1:5, ]

top_5_wt_IGF3_I474M <- top_5_wt_IGF3_I474M %>%
    mutate(gene_list = strsplit(as.character(geneID), "/"))

avg_fc_list <- list()



for (i in seq_len(nrow(top_5_wt_IGF3_I474M))) {
    genes <- unlist(top_5_wt_IGF3_I474M$gene_list[i])
    IDs <- mapIds(x = org.Hs.eg.db, keys = genes, column = "ENSEMBL", keytype
= "ENTREZID")
    # Step 4: Filter IGF2BP2a_diff_bound for matching genes and get log2FC
    matching_fc <- IGF2BP3wt_diff_bound %>%
        filter(gene_ID %in% IDs) %>%
        pull(log2FC)  # Extract fold change values

    # Compute average log2FC
    avg_fc <- ifelse(length(matching_fc) > 0, mean(matching_fc, na.rm =
TRUE), NA)

    avg_fc_list[[i]] <- avg_fc
}



top_5_wt_IGF3_I474M$avg_log2FC <- unlist(avg_fc_list)

FigIGF3_I474M_wt <- ggplot(data = top_5_wt_IGF3_I474M, aes(x = Description, y
= avg_log2FC)) +
    geom_bar(stat = "identity") + coord_flip() + scale_alpha_binned(range =
c(0.5,
    1)) + ylim(-3, 3) + theme_pubr(legend = "left")
```

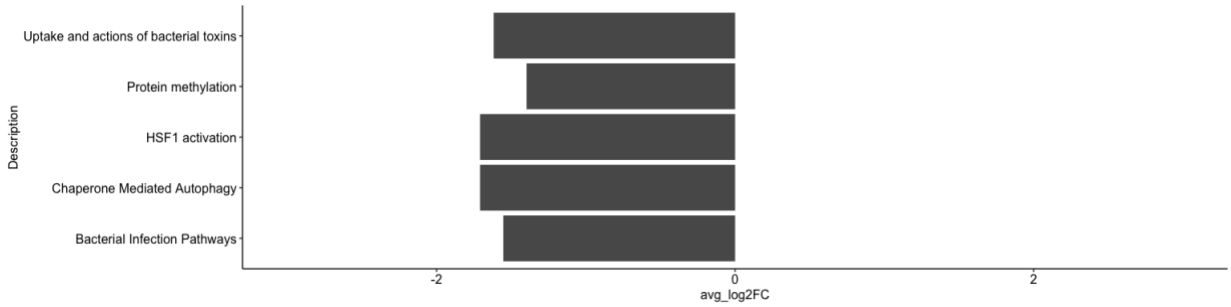
```
FigIGF3_I474M_wt
```




```
IDs_I474M <- mapIds(x = org.Hs.eg.db, keys =
IGF2BP3WTvsI474M_diff_bound_gene_ID$gene_ID,
    column = "ENTREZID", keytype = "ENSEMBL")
IDs_I474M <- as.character(IDs_I474M)[is.na(as.character(IDs_I474M)) == FALSE]

# Pathway analysis
Reactome_I474M <- enrichPathway(gene = IDs_I474M, organism = "human",
pAdjustMethod = "BH",
    universe = IDs_background, readable = FALSE)


# IGF2BP3 WTvsR525C
IGF2BP3wt_diff_bound <- WTvsR525C[WTvsR525C$DiffBind == "wt", ]
IGF2BP3WTvsR525C_diff_bound <- WTvsR525C[WTvsR525C$DiffBind == "R525C", ]



IGF2BP3wt_diff_bound <- IGF2BP3wt_diff_bound %>%
    mutate(gene_ID = strsplit(gene_ID, "\\.") %>%
        lapply(., function(x) x[1]) %>%
        unlist())

IGF2BP3WTvsR525C_diff_bound <- IGF2BP3WTvsR525C_diff_bound %>%
    mutate(gene_ID = strsplit(gene_ID, "\\.") %>%
        lapply(., function(x) x[1]) %>%
        unlist())

IGF2BP3wt_diff_bound_gene_ID <- subset(IGF2BP3wt_diff_bound, select = c(18))
IGF2BP3WTvsR525C_diff_bound_gene_ID <- subset(IGF2BP3WTvsR525C_diff_bound,
select = c(18))



IDs_wt <- mapIds(x = org.Hs.eg.db, keys =
IGF2BP3wt_diff_bound_gene_ID$gene_ID, column = "ENTREZID",
    keytype = "ENSEMBL")
```

```r
IDs_wt <- as.character(IDs_wt)[is.na(as.character(IDs_wt)) == FALSE]
IDs_background <- mapIds(x = org.Hs.eg.db, keys = Inputs$Geneid, column =
"ENTREZID",
    keytype = "ENSEMBL")
IDs_background <-
as.character(IDs_background)[is.na(as.character(IDs_background)) ==
    FALSE]

# Pathway analysis IGF2BP3 wt
Reactome_wt_IGF3_R525C <- enrichPathway(gene = IDs_wt, organism = "human",
pAdjustMethod = "BH",
    universe = IDs_background, readable = FALSE)


Pathways_wt_IGF3_R525C <- as.data.frame(Reactome_wt_IGF3_R525C)


top_5_wt_IGF3_R525C <- Pathways_wt_IGF3_R525C[1:5, ]

top_5_wt_IGF3_R525C <- top_5_wt_IGF3_R525C %>%
    mutate(gene_list = strsplit(as.character(geneID), "/"))

avg_fc_list <- list()


for (i in seq_len(nrow(top_5_wt_IGF3_R525C))) {
    genes <- unlist(top_5_wt_IGF3_R525C$gene_list[i])
    IDs <- mapIds(x = org.Hs.eg.db, keys = genes, column = "ENSEMBL", keytype
= "ENTREZID")
    # Step 4: Filter IGF2BP2a_diff_bound for matching genes and get log2FC
    matching_fc <- IGF2BP3wt_diff_bound %>%
        filter(gene_ID %in% IDs) %>%
        pull(log2FC)  # Extract fold change values

    # Compute average log2FC
    avg_fc <- ifelse(length(matching_fc) > 0, mean(matching_fc, na.rm =
TRUE), NA)

    avg_fc_list[[i]] <- avg_fc
}


top_5_wt_IGF3_R525C$avg_log2FC <- unlist(avg_fc_list)

FigIGF3_R525C_wt <- ggplot(data = top_5_wt_IGF3_R525C, aes(x = Description, y
= avg_log2FC)) +
    geom_bar(stat = "identity") + coord_flip() + scale_alpha_binned(range =
c(0.5,
```

```
    1)) + ylim(-3, 3) + theme_pubr(legend = "left")

FigIGF3_R525C_wt
```




```
IDs_R525C <- mapIds(x = org.Hs.eg.db, keys =
IGF2BP3WTvsR525C_diff_bound_gene_ID$gene_ID,
    column = "ENTREZID", keytype = "ENSEMBL")
IDs_R525C <- as.character(IDs_R525C)[is.na(as.character(IDs_R525C)) == FALSE]

# Pathway analysis IGF2BP1R167H
Reactome_R525C <- enrichPathway(gene = IDs_R525C, organism = "human",
pAdjustMethod = "BH",
    universe = IDs_background, readable = FALSE)
```

Figure S5A

```
HOM_list <- read.table(file =
"/Users/riccardomosca/Desktop/RAPseq_PAPER/FIGUREs/FIGURE7/SOFIA_mouse_data/H
OM_MouseHuman.txt",
    header = T)

IGFs_hs <- HOM_list[HOM_list$Symbol %in% c("IGF2BP1", "IGF2BP2", "IGF2BP3"),
8]
IGFs_mm <- HOM_list[HOM_list$Symbol %in% c("IGF2BP1", "IGF2BP2", "IGF2BP3"),
9]

# File with the RPKM values from the Cardaso et al study downloaded from
# https://apps.kaessmannlab.org/evodevoapp/
human_rpkm <- read.table(file =
"/Users/riccardomosca/Desktop/RAPseq_PAPER/FIGUREs/FIGURE7/SOFIA_mouse_data/H
uman_rpkm.txt",
    header = T, row.names = 1)
human_rpkm <- human_rpkm[row.names(human_rpkm) %in% IGFs_hs, ]
Hs_Liver <- human_rpkm[, str_starts(colnames(human_rpkm), "Liver") == TRUE]
gene_names <- row.names(Hs_Liver)
Hs_Liver <- as.data.frame(lapply(Hs_Liver, function(x)
as.numeric(as.character(x))))
row.names(Hs_Liver) <- gene_names

# Mean the RPKM of each technical replicate
```

```r
Hs_Liver <- as.data.frame(t(Hs_Liver))
group <- sapply(str_split(as.character(row.names(Hs_Liver)), "\\."), `[`, 2)
Hs_Liver <- cbind(group, Hs_Liver)
Hs_Liver <- Hs_Liver %>%
    group_by(group) %>%
    summarise_all(funs(as.numeric(mean(., na.rm = TRUE))))

## Birth = timepoint between 13 and 14
human_stages <-
read.table("/Users/riccardomosca/Desktop/RAPseq_PAPER/FIGUREs/FIGURE7/SOFIA_mouse_data/Human_stages_key.txt")
names(human_stages) <- c("group", "stage")
Hs_Liver <- merge(human_stages, Hs_Liver, by = "group")
row.names(Hs_Liver) <- paste(Hs_Liver$stage, Hs_Liver$group, sep = "_")
Hs_Liver <- Hs_Liver[, -c(1, 2)]
Hs_Liver <- as.data.frame(t(Hs_Liver))
Hs_Liver$gene <- row.names(Hs_Liver)

plotting_Hs <- tidyr::gather(Hs_Liver, "Developmental_Stage", "expression", -c(gene))
plotting_Hs$Stage <-
sapply(str_split(as.character(plotting_Hs$Developmental_Stage),
    "_"), `[`, 1)
plotting_Hs$Developmental_Stage <-
sapply(str_split(as.character(plotting_Hs$Developmental_Stage),
    "_"), `[`, 2)
plotting_Hs$Study <- rep("Cardosso", nrow(plotting_Hs))
plotting_Hs$species <- rep("H.sapiens", nrow(plotting_Hs))

# File with the RPKM values from the Cardaso et al study downloaded from
# https://apps.kaessmannlab.org/evodevoapp/
mouse_rpkm <-
read.table("/Users/riccardomosca/Desktop/RAPseq_PAPER/FIGUREs/FIGURE7/SOFIA_mouse_data/Mouse_rpkm.txt",
    header = T, row.names = 1)
mouse_rpkm <- mouse_rpkm[row.names(mouse_rpkm) %in% IGFs_mm, ]
Mm_Liver <- mouse_rpkm[, str_starts(colnames(mouse_rpkm), "Liver") == TRUE]
gene_names <- row.names(Mm_Liver)
Mm_Liver <- as.data.frame(lapply(Mm_Liver, function(x)
as.numeric(as.character(x))))
row.names(Mm_Liver) <- gene_names

# Mean the RPKM of each technical replicate
Mm_Liver <- as.data.frame(t(Mm_Liver))
group <- sapply(str_split(as.character(row.names(Mm_Liver)), "\\."), `[`, 2)
Mm_Liver <- cbind(group, Mm_Liver)
Mm_Liver <- Mm_Liver %>%
    group_by(group) %>%
    summarise_all(funs(as.numeric(mean(., na.rm = TRUE))))
```

```
mouse_stages <-
read.table("/Users/riccardomosca/Desktop/RAPseq_PAPER/FIGUREs/FIGURE7/SOFIA_m
ouse_data/Mouse_stages_key.txt")
names(mouse_stages) <- c("group", "stage")
Mm_Liver <- merge(mouse_stages, Mm_Liver, by = "group")
row.names(Mm_Liver) <- paste(Mm_Liver$stage, Mm_Liver$group, sep = "_")
Mm_Liver <- Mm_Liver[, -c(1, 2)]
Mm_Liver <- as.data.frame(t(Mm_Liver))
Mm_Liver$gene <- row.names(Mm_Liver)
plotting_Mm <- tidyr::gather(Mm_Liver, "Developmental_Stage", "expression", -
c(gene))
plotting_Mm$Stage <-
sapply(str_split(as.character(plotting_Mm$Developmental_Stage),
    "_"), `[`, 1)
plotting_Mm$Developmental_Stage <-
sapply(str_split(as.character(plotting_Mm$Developmental_Stage),
    "_"), `[`, 2)
plotting_Mm$Study <- rep("Cardosso", nrow(plotting_Mm))
plotting_Mm$species <- rep("M.musculus", nrow(plotting_Mm))

# Function for plotting human and mouse develomental stages
making_plots_S5A <- function(gene_list) {
    colors <- c("#08306B", "#4292C6", "#DEEBF7")
    genes <- gene_list
    mouse_gene <- HOM_list[HOM_list$homo_ensembl %in% genes, "mouse_ensembl"]
    data_hs <- merge(plotting_Hs[plotting_Hs$gene %in% genes, ], HOM_list[,
c(7,
        8)], by.x = 1, by.y = 2)
    hs_data <- arrange(data_hs, Symbol)
    hs_data$Developmental_Stage <- factor(hs_data$Developmental_Stage, levels
= unique(arrange(hs_data,
        as.numeric(Stage))[, 2]))
    data_mm <- merge(plotting_Mm[plotting_Mm$gene %in% mouse_gene, ],
HOM_list[,
        c(4, 9)], by.x = 1, by.y = 2)
    mm_data <- arrange(data_mm, mouse_gene)
    mm_data$Developmental_Stage <- factor(mm_data$Developmental_Stage, levels
= unique(arrange(mm_data,
        as.numeric(Stage))[, 2]))

    Hs_dev <- ggplot(data = hs_data) + geom_line(aes(x = Developmental_Stage,
y = (as.numeric(expression)),
        group = Symbol, col = Symbol), size = 2, alpha = 0.75) +
geom_point(aes(x = Developmental_Stage,
        y = (as.numeric(expression)), fill = Symbol), size = 5, shape = 21) +
geom_vline(xintercept = 13.5,
        linetype = 2, color = "gray", size = 0.8) + scale_color_manual(values
= colors) +
```

```
        xlab("Developmental Stage") + ylab("RPKM") + ggtitle("Homo_sapiens")
+ ylim(0,
        80) + theme_pubr() + scale_fill_manual(values = colors) +
scale_color_manual(values = colors)
    Hs_dev <- Hs_dev + theme(axis.text.x = element_text(angle = 90, vjust =
0.5,
        hjust = 1), legend.position = "bottom")

    Mm_dev <- ggplot(data = mm_data) + geom_line(aes(x = Developmental_Stage,
y = (as.numeric(expression)),
        group = mouse_gene, col = mouse_gene), size = 2, alpha = 0.75) +
geom_point(aes(x = Developmental_Stage,
        y = (as.numeric(expression)), fill = mouse_gene), size = 5, shape =
21) +
        geom_vline(xintercept = 9.5, linetype = 2, color = "gray", size =
0.8) +
        scale_color_manual(values = colors) + xlab("Developmental Stage") +
ylab("RPKM") +
        ggtitle("Mus musculus") + ylim(0, 80) + theme_pubr() +
scale_fill_manual(values = colors) +
        scale_color_manual(values = colors)
    Mm_dev <- Mm_dev + theme(axis.text.x = element_text(angle = 90, vjust =
0.5,
        hjust = 1), legend.position = "bottom")

    gene_plots <- ggarrange(Hs_dev, Mm_dev, ncol = 2, align = "hv")

    return(gene_plots)
}


making_plots_S5A(IGFs_hs)
```

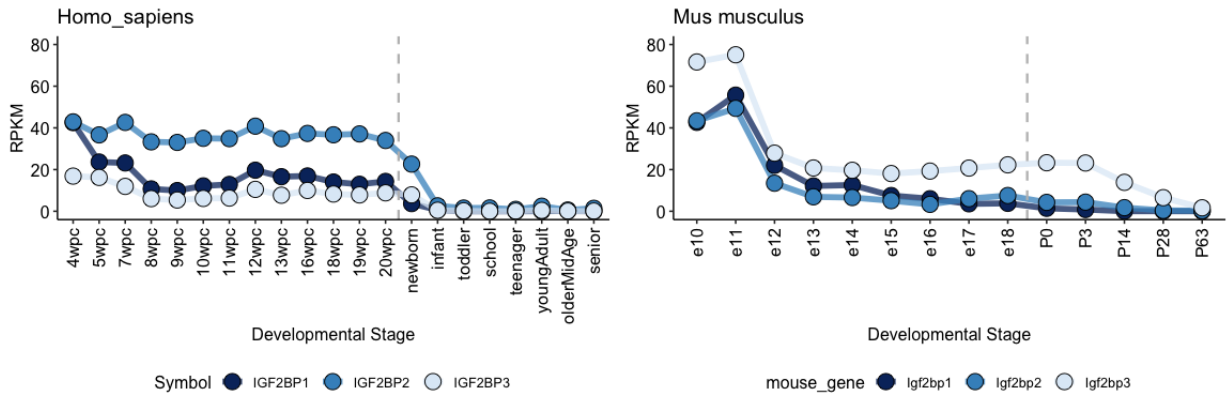


Figure S5B

```
# Downoladed table of normalized counts from the Schmitt et al study PLOS ONE
# Human liver cancer cell lines
```

```
human_cell_lines <-
read.table("/Users/riccardomosca/Desktop/RAPseq_PAPER/FIGUREs/FIGURE7/SOFIA_m
ouse_data/journal.pgen.1006024.s014.TSV",
    header = T)
human_cell_lines <- human_cell_lines[, c(1, 3:5)]
human_cell_lines <- gather(human_cell_lines, "Sample", "Normalized_count",
2:4)
human_cell_lines <- human_cell_lines[human_cell_lines$Gene %in% IGFs_hs, ]

# Mouse liver cancer cell lines
mouse_cell_lines <-
read.table("/Users/riccardomosca/Desktop/RAPseq_PAPER/FIGUREs/FIGURE7/SOFIA_m
ouse_data/journal.pgen.1006024.s015.TSV",
    header = T)
mouse_cell_lines <- mouse_cell_lines[, c(1, 4:6)]
mouse_cell_lines <- gather(mouse_cell_lines, "Sample", "Normalized_count",
2:4)
mouse_cell_lines <- mouse_cell_lines[mouse_cell_lines$Gene %in% IGFs_mm, ]

# Function for plotting human and mouse liver cancer cell lines
making_plots_S5B <- function(x) {
    colors <- c("#08306B", "#4292C6", "#DEEBF7")
    human_cell_lines <- merge(human_cell_lines, HOM_list[, c(7, 8)], by.x =
1, by.y = 2)
    mouse_cell_lines <- merge(mouse_cell_lines, HOM_list[, c(4, 9)], by.x =
1, by.y = 2)

    cancer_Hs <- ggplot(human_cell_lines) + geom_bar(aes(x = Sample, y =
Normalized_count,
        color = Symbol, fill = Symbol), stat = "identity", position =
position_dodge(),
        width = 0.75) + xlab(NULL) + ylab("Normalized Counts") +
scale_color_manual(values = colors) +
        scale_fill_manual(values = colors) + ggtitle("Human cell lines") +
theme_pubr() +
        ylim(0, 250)
    cancer_Hs <- cancer_Hs + theme(legend.position = "bottom")

    cancer_Mm <- ggplot(mouse_cell_lines) + geom_bar(aes(x = Sample, y =
Normalized_count,
        color = mouse_gene, fill = mouse_gene), stat = "identity", position =
position_dodge(),
        width = 0.75) + xlab(NULL) + ylab("Normalized Counts") +
scale_color_manual(values = colors) +
        scale_fill_manual(values = colors) + ggtitle("Mouse cell lines") +
theme_pubr() +
        ylim(0, 250)
    cancer_Mm <- cancer_Mm + theme(legend.position = "bottom")
```

```
    cancer_plots <- ggarrange(cancer_Hs, cancer_Mm, ncol = 2, align = "hv")
    return(cancer_plots)
}

making_plots_S5B()
```

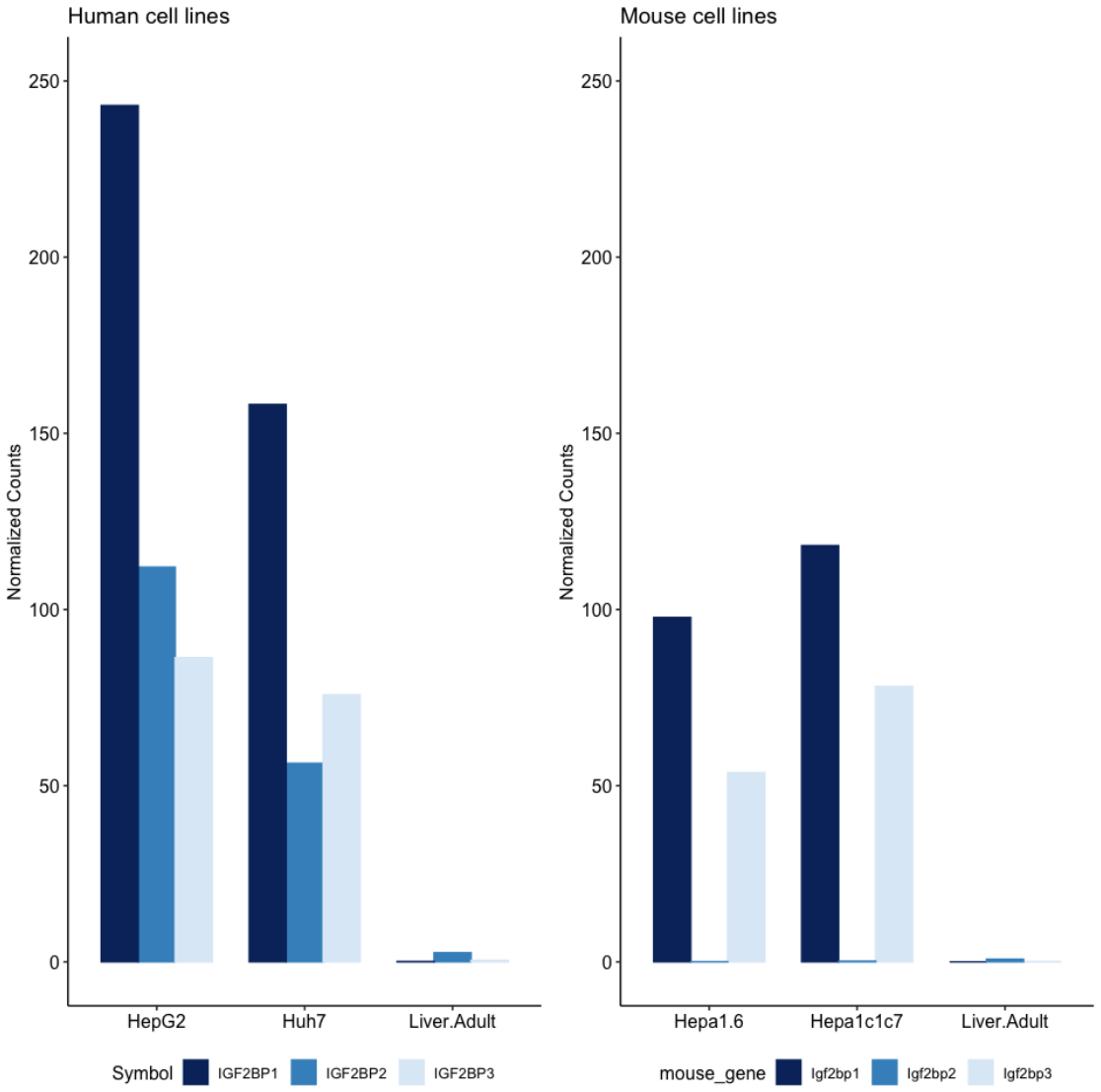


Figure S5D

```
PATH <-
"/Users/riccardomosca/Desktop/RAPseq_PAPER/PEAKs/ANNOTATED/IGFs_Fig7/"
PEAKS <- list.files(path = PATH)
Names <- unlist(str_split(PEAKS, "\\_final"))[grep("txt",
unlist(str_split(PEAKS,
    "\\_final")), invert = T)]
Tables <- list()
for (i in 1:length(Names)) {
    Tables[[i]] <- as.data.frame(read.table(paste(PATH, PEAKS[i], sep = ""),
```

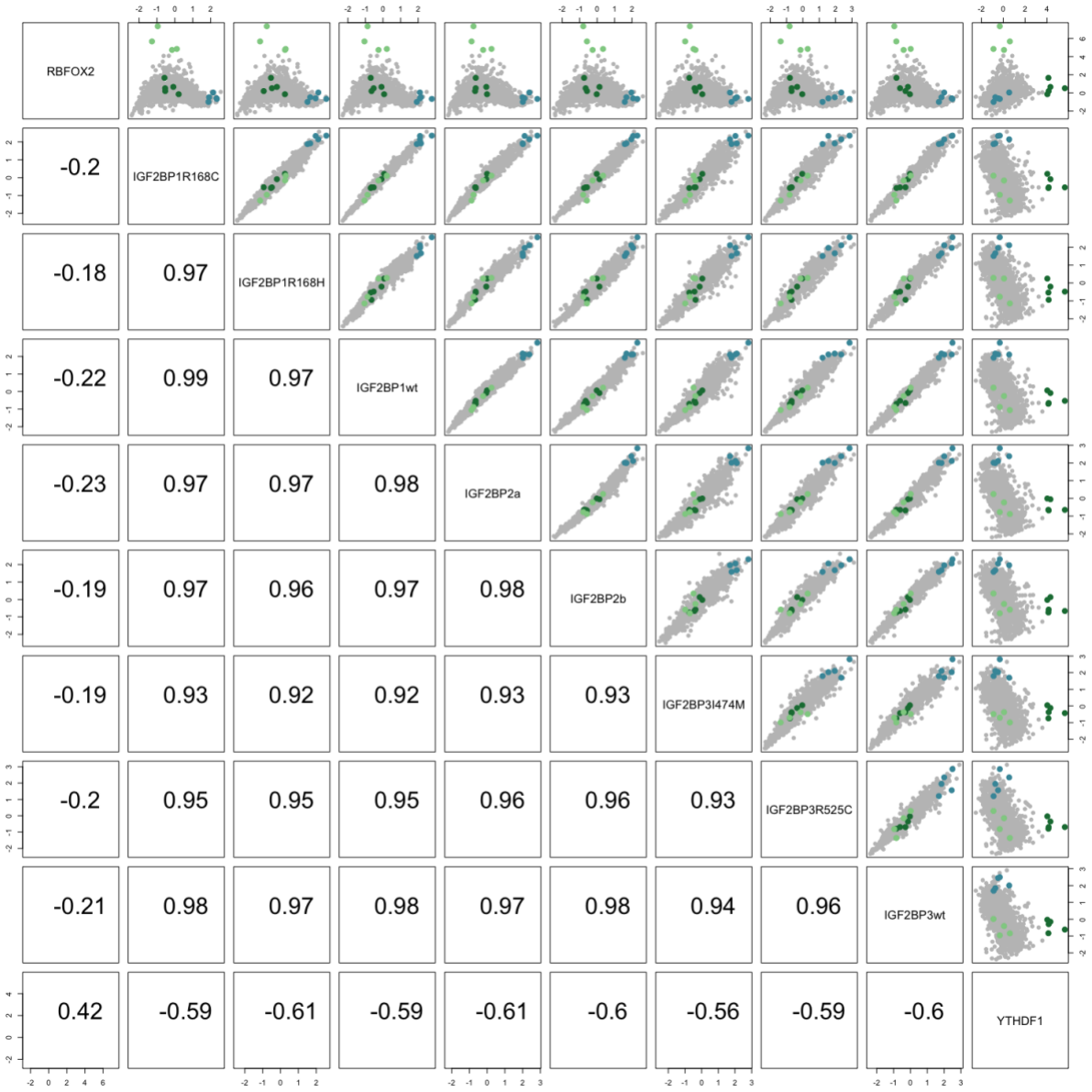
```
stringsAsFactors = F,
        header = T))
    Tables[[i]]$positive_fa_sub <- str_sub(Tables[[i]]$positive_fa, 50, 150)
}
names(Tables) <- Names
Names

##  [1] "IGF2BP1R168C" "IGF2BP1R168H" "IGF2BP1wt"    "IGF2BP2a"
"IGF2BP2b"
##  [6] "IGF2BP3I474M" "IGF2BP3R525C" "IGF2BP3wt"    "RBFOX2"       "YTHDF1"

kmer_table <- as.data.frame(Names)
STRINGS <- c()
for (i in Names) {
    STRINGS <- c(STRINGS, as.character(paste(Tables[[i]]$positive_fa_sub,
collapse = "NN")))
}
k = 5
kmer_table <- cbind(kmer_table, STRINGS)
kmer_table[] <- lapply(kmer_table, as.character)
bases <- c("A", "C", "T", "G")
kmers <- unite(as.data.frame(permutations(n = 4, r = k, v = bases,
repeats.allowed = T)),
    col = kmers, sep = "")
for (i in 1:nrow(kmer_table)) {
    kmers <- cbind(kmers, str_count(kmer_table[i, 2], kmers[,
1])/length(kmer_table[i,
        2]))
}
rownames(kmers) <- kmers$kmers
pos_kmers_5 <- as.data.frame(kmers[, -1], row.names = rownames(kmers))
colnames(pos_kmers_5) <- kmer_table$Names









# All the binding sites from all the RBPs profiled in the Manuscript are used
# as background for the normalization and Z transformation
PATH <- "/Users/riccardomosca/Desktop/RAPseq_PAPER/PEAKs/ANNOTATED/ALL/"
PEAKS <- list.files(path = PATH)
Names <- unlist(str_split(PEAKS, "\\_final"))[grep("txt",
unlist(str_split(PEAKS,
    "\\_final")), invert = T)]
Tables <- list()
for (i in 1:length(Names)) {
```

```
    Tables[[i]] <- as.data.frame(read.table(paste(PATH, PEAKS[i], sep = ""),
stringsAsFactors = F,
        header = T))
    Tables[[i]]$negative_fa_sub <- str_sub(Tables[[i]]$positive_fa, 50, 150)
}
names(Tables) <- Names
Names
```

```
##  [1] "ABCE1"      "BRD2"       "BTF3"       "CCDC124"    "CCDC59"
##  [6] "CCDC86"     "CCT2"       "CCT3"       "DNAJA1"     "ENO1"
## [11] "FAM98A"     "FSCN1"      "ggHuRHUMAN" "hnRNPA1"    "hnRNPC"
## [16] "hsHuRHUMAN" "HSPA5"      "HSPA8"      "HSPA9"      "HuRPTBP1"
## [21] "IGF2BP1wt"  "IGF2BP2a"   "IGF2BP3wt"  "IRP1"       "MANF"
## [26] "MAPRE1"     "mdHuRHUMAN" "mmHuRHUMAN" "NAP1L1"     "NASP"
## [31] "PEBP1"      "PKM"        "PRDX6"      "PRMT1"      "PTBP1"
## [36] "RAN"        "RBFOX2"     "STMN1"      "TKT"        "xtHuRHUMAN"
## [41] "YBX3"       "YTHDF1"
```

```
kmer_table <- as.data.frame(Names)
STRINGS <- c()
for (i in Names) {
    STRINGS <- c(STRINGS, as.character(paste(Tables[[i]]$negative_fa_sub,
collapse = "NN")))
}
k = 5
kmer_table <- cbind(kmer_table, STRINGS)
kmer_table[] <- lapply(kmer_table, as.character)
bases <- c("A", "C", "T", "G")
kmers <- unite(as.data.frame(permutations(n = 4, r = k, v = bases,
repeats.allowed = T)),
    col = kmers, sep = "")
for (i in 1:nrow(kmer_table)) {
    kmers <- cbind(kmers, str_count(kmer_table[i, 2], kmers[,
1])/length(kmer_table[i,
        2]))
}
rownames(kmers) <- kmers$kmers
neg_kmers_5 <- as.data.frame(kmers[, -1], row.names = rownames(kmers))
colnames(neg_kmers_5) <- kmer_table$Names



# Normalization and Z transformation
Z_k5 <-
as.data.frame(scale(t((t(pos_kmers_5)/colSums(pos_kmers_5)))/rowMeans(t(t(neg
_kmers_5)/colSums(neg_kmers_5)))))



Mots_YTH <- Z_k5[grep("GGACT|TGGAC|GACTG|GACTC|CGGAC", rownames(Z_k5)), ]
Mots_YTH$COLOR <- rep("#197B41", nrow(Mots_YTH))
```

```r
Mots_YTH$SIZE <- rep(1.5, nrow(Mots_YTH))
Mots_IGFs <- Z_k5[grep("ACAAC|CAAAC|AACAC|CACAA|CAACA", rownames(Z_k5)), ]
Mots_IGFs$COLOR <- rep("#4397A8", nrow(Mots_IGFs))
Mots_IGFs$SIZE <- rep(1.5, nrow(Mots_IGFs))
Mots_RBF <- Z_k5[grep("GCATG|TGCAT|GAATG|GCACG", rownames(Z_k5)), ]
Mots_RBF$COLOR <- rep("#8FCF91", nrow(Mots_RBF))
Mots_RBF$SIZE <- rep(1.5, nrow(Mots_RBF))
Kmers <-
Z_k5[grep("CGGAC|GGACT|TGGAC|GACTG|GACTC|ACAAC|CAAAC|AACAC|CACAA|CAACA|GCATG|
TGCAT|GAATG|GCACG",
    rownames(Z_k5), invert = T), ]
Kmers$COLOR <- rep("#C0C0C0", nrow(Kmers))
Kmers$SIZE <- rep(1, nrow(Kmers))
Z_k5_bis <- rbind(Kmers, Mots_IGFs, Mots_YTH, Mots_RBF)
# Correlation panel
panel.cor <- function(x, y) {
    usr <- par("usr")
    on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    rho <- round(cor(x, y, method = "spearman"), digits = 2)
    txt <- as.character(rho)
    cex.cor <- 0.8/strwidth(txt)
    text(0.6, 0.6, txt, cex = 3)
}
# Customize upper panel
upper.panel <- function(x, y) {
    points(x, y, pch = 19, col = Z_k5_bis$COLOR, cex = Z_k5_bis$SIZE)
}


cor_panel <- pairs(Z_k5_bis[, c(9, 1:8, 10)], lower.panel = panel.cor,
upper.panel = upper.panel)
```


RBFOX2
IGF2BP1R168C
IGF2BP1R168H
IGF2BP1wt
IGF2BP2a
IGF2BP2b
IGF2BP3I474M
IGF2BP3R525C
IGF2BP3wt
YTHDF1
-0.2
-0.18 0.97
-0.22 0.99 0.97
-0.23 0.97 0.97 0.98
-0.19 0.97 0.96 0.97 0.98
-0.19 0.93 0.92 0.92 0.93 0.93
-0.2 0.95 0.95 0.95 0.96 0.96 0.93
-0.21 0.98 0.97 0.98 0.97 0.98 0.94 0.96
0.42 -0.59 -0.61 -0.59 -0.61 -0.6 -0.56 -0.59 -0.6