Differential binding of AGO in mir181KO vs WT

Mirko Brüggemann, Melina Klostermann

14 April, 2023

Contents

1 Libraries and settings **1**

2 What was done? **3**

3 Combine binding sites **3**

4 Perform Differntial binding **4**

5 Results (after Gene Expression filter) **14**

6 Session Info **27**

1 Libraries and settings

```
# ----------------------------------------
# libraries
# ----------------------------------------

library(rtracklayer)
library(GenomicRanges)
library(ggplot2)
library(AnnotationDbi)
library(dplyr)
library(reshape2)
library(UpSetR)
library(GenomicFeatures)
library(kableExtra)
library(knitr)
library(ggrepel)
library(gridExtra)
library(grid)
library(viridis)
library(BindingSiteFinder)
library(ComplexHeatmap)
library(forcats)
library(ggtext)
library(patchwork)
library(tibble)
library(tidyr)
library(dplyr)
```

```
library(ggpointdensity)
library(ggsci)
library(ggtext)
library(ggrepel)
library(patchwork)
library(ggrastr)
library(matrixStats)
library(DESeq2)
library(IHW)


source("/Users/melinaklostermann/Documents/projects/AgoCLIP_miR181/mirko_files/mirECLIP/DifferentialBin
source("/Users/melinaklostermann/Documents/projects/R_general_functions/theme_paper.R")
source("/Users/melinaklostermann/Documents/projects/AgoCLIP_miR181/mirko_files/mirECLIP/DifferentialBin

# ----------------------------------------
# settings
# ----------------------------------------

out <- "/Users/melinaklostermann/Documents/projects/AgoCLIP_miR181/R_github/miR181_paper/Figure1/Differ
tpm_cut <- 50
```

```
# ----------------------------------------
# ----------------------------------------
# Files
# ----------------------------------------
# ----------------------------------------

# ----------------------------------------
# annotation
# ----------------------------------------

annoDb <- loadDb("/Users/melinaklostermann/Documents/projects/AgoCLIP_miR181/R_github/miR181_paper/Meth
gns <- readRDS("/Users/melinaklostermann/Documents/projects/AgoCLIP_miR181/R_github/miR181_paper/Method

# ----------------------------------------
# AGO binding sites
# ----------------------------------------

bs_wt <- readRDS("/Users/melinaklostermann/Documents/projects/AgoCLIP_miR181/R_github/miR181_paper/Meth

bs_ko <- readRDS("/Users/melinaklostermann/Documents/projects/AgoCLIP_miR181/R_github/miR181_paper/Meth

# ----------------------------------------
# Load clip data
# ----------------------------------------

clipFiles = "/Users/melinaklostermann/Documents/projects/AgoCLIP_miR181/pipe_output_22_02_14/non-chimer
clipFiles = list.files(clipFiles, pattern = ".bw$", full.names = TRUE)
clipFiles = clipFiles[!grepl("Inp", clipFiles)]
clipFiles = clipFiles[!grepl("miR181_", clipFiles)]
clipFilesP = clipFiles[grep(clipFiles, pattern = "plus")]
clipFilesM = clipFiles[grep(clipFiles, pattern = "minus")]
```

Table 1: Merge and combine

Option	nRanges
inputRanges	37,575
mergeCrosslinkSites	28,346
minCrosslinks	28,346
minClSites	28,346
centerIsClSite	28,253
centerIsSummit	NA

2 What was done?

3 Combine binding sites

Binding sites from both conditions do either overlap exactly, overlap partially or don't overlap at all. In the following partial overlaps are resolved by re-centering the binding sites based on the highest crosslink signal of both conditions.

```
# merge BS from both conditions
bsMerge = unique(c(resize(bs_wt, fix = "center", width = 1),
                   resize(bs_ko, fix = "center", width = 1)))

# Organize clip data in dataframe
colData = data.frame(
  id = c(1:6),
  condition = factor(c(rep("KO",3), rep("WT",3)),
                     levels = c("KO", "WT")),
                     clPlus = clipFilesP,
                     clMinus = clipFilesM)

# Make BindingSiteFinder object
bds = BSFDataSetFromBigWig(ranges = bsMerge, meta = colData)


# Make unified binding sites from both conditions
bdsMerge <- makeBindingSites(object = bds, bsSize = 7, minWidth = 0,
                         minCrosslinks = 0, minClSites = 0, centerIsSummit = FALSE)

df = getSummary(bdsMerge)
kable(myNumberFormat(df), caption = "Merge and combine")
```

Combination of both sets of binding sites results in a single set (N=28,253), where each binding site was either seen in *WT*, *KO* or both conditions. The figure below shows the set sizes in more detail.

Annotate combined binding sites with *PureCLIP score*, *WT* and *KO* status information.

```
# annotate with pureclip score
rng = getRanges(bds)
rng$additionalScores = NULL
mcols(rng)$score = rng$scoreSum
bdsFinal = annotateWithScore(bdsMerge, rng)

# annotate with condition support
rngFinal = getRanges(bdsFinal)
```

```
r1 = resize(bs_wt, fix = "center", width = 1)
r2 = resize(bs_ko, fix = "center", width = 1)
condition_support = data.frame(WT = countOverlaps(rngFinal, r1),
                KO = countOverlaps(rngFinal, r2))
mcols(rngFinal) = cbind(mcols(rngFinal), condition_support)

# transfer the geneID
olsWT = findOverlaps(rngFinal, bs_wt)
olsKO = findOverlaps(rngFinal, bs_ko)
rngFinal$geneID = NA
rngFinal$geneID[queryHits(olsWT)] = bs_wt$geneID[subjectHits(olsWT)]
rngFinal$geneID[queryHits(olsKO)] = bs_ko$geneID[subjectHits(olsKO)]

# transfer gene names
rngFinal$geneName = NA
rngFinal$geneName[queryHits(olsWT)] = bs_wt$geneName[subjectHits(olsWT)]
rngFinal$geneName[queryHits(olsKO)] = bs_ko$geneName[subjectHits(olsKO)]

# transfer transcript region
rngFinal$region = NA
rngFinal$region[queryHits(olsWT)] = bs_wt$region[subjectHits(olsWT)]
rngFinal$region[queryHits(olsKO)] = bs_ko$region[subjectHits(olsKO)]

# set final range
bdsFinal = setRanges(bdsFinal, rngFinal)
```

4 Perform Differntial binding

In order to perform differential binding analysis, we use DEseq2. The design formula contains both the number of crosslinks per binding site and the background crosslinks in the whole gene of the binding site.

4.1 Calculate crosslinks in bs and in gene background

The number of background crosslinks per gene is calculated to infer upregulation or downregulation of genes between the two conditions. The background contains all crosslinks on a gene except crosslinks in a binding site or within a 5nt offset to the binding sites.

```
### ============================================================================
### Get crosslink numbers in background and binding sites
### ----------------------------------------------------------------------------
# Function get coverage per bs
#-------------------------
# NOTE: this is a simple copy from the BindingSiteFinder package, that uses
# the range and signal objects directly as input.
coverageBySignal <- function(range, signal,
         merge = TRUE,
         returnType = c("GRanges", "matrix", "data.frame")) {

    # split by strand
    rng = range
    rngPlus = rng[strand(rng) == "+"]
    rngMinus = rng[strand(rng) == "-"]
    # prepare signal
```

```
    sgn = signal

    # signal coverage is reported for each position in the range of the peak
    if (!isTRUE(merge)) {
        # manage return type
        # only return type data.frame is possible with this option
        returnType = match.arg(returnType,
                                choices = c("GRanges", "matrix", "data.frame"))
        if (returnType != "data.frame") {
            warning("Only return type 'data.frame' possible with non-merged output.")
        }
        returnType = "data.frame"

        if (length(rngPlus) > 0) {
            matPlus = lapply(sgn$signalPlus, function(x) {
                as.matrix(x[rngPlus])
            })
            covPlus = do.call(rbind, lapply(matPlus, colSums))
        }
        if (length(rngPlus) == 0) {
            covPlus = 0
        }
        if (length(rngMinus) > 0) {
            matMinus = lapply(sgn$signalMinus, function(x) {
                as.matrix(x[rngMinus])
            })
            covMinus = do.call(rbind, lapply(matMinus, colSums))
            # flip orientation of minus strand coverage
            covMinus = covMinus %>% as.data.frame() %>% rev() %>% as.matrix()
        }
        if (length(rngMinus) == 0) {
            covMinus = 0
        }
        covDf = covPlus + covMinus
        retObj = as.data.frame(covDf)
    }
    # signal is merged over all positions in the range
    if (isTRUE(merge)) {
        mcols(rngPlus) = as.matrix(
            do.call(cbind, lapply(sgn$signalPlus, function(x) {
                sum(x[rngPlus])
            })))
        mcols(rngMinus) = as.matrix(
            do.call(cbind, lapply(sgn$signalMinus, function(x) {
                sum(x[rngMinus])
            })))
        # sort ranges
        rngCov = c(rngPlus, rngMinus)
        rngCov = GenomeInfoDb::sortSeqlevels(rngCov)
        rngCov = sort(rngCov)
        # manage return type
        returnType = match.arg(returnType,
                                choices = c("GRanges", "matrix", "data.frame"))
```

```
        if (returnType == "GRanges") {
            retObj = rngCov
        }
        if (returnType == "matrix") {
            retObj = as.matrix(mcols(rngCov))
        }
        if (returnType == "data.frame") {
            retObj = as.data.frame(mcols(rngCov))
        }
    }
    return(retObj)
}

# Make Matrix from background Signal
#------------------------
makeBsBackgroundMatrix <- function(geneRanges, object, offset, matchBy = "geneID"){
    # reassign input
    genes = geneRanges
    bs = getRanges(object)
    bsWidth = unique(width(bs))
    signal = getSignal(object)

        # prepare the background region
        gen = genes[genes$geneID %in% bs$geneID]
        gen = sort(sortSeqlevels(gen))
        gen = as(gen, "GRangesList")
        # group binding sites per gene
        bs = sort(sortSeqlevels(bs))
        bsNames = mcols(bs) %>%
            as.data.frame() %>%
            group_by(geneID) %>%
            mutate(name = paste0("bs",seq_along(geneID))) %>%
            pull(name)
        names(bs) = bsNames
        bs = split(bs, bs$geneID)
        idx = match(names(gen), names(bs))
        bs = bs[idx]

    # add a protective range around each binding site
    # bsOffset = bs + offset
    bsOffset = resize(bs, fix = "center", width = bsWidth + offset)

    # remove binding sites ranges from gene ranges to create background
    bac = GenomicRanges::disjoin(pc(gen, bsOffset), with.revmap = TRUE)
    bac = unlist(bac)
    len = lapply(bac$revmap, length)
    bac = bac[len == 1]
    bac = split(bac, names(bac))
    bac = unlist(bac, use.names = F)

    # count crosslinks in background regions summarized by gene
    bac = coverageBySignal(range = bac, signal = signal, returnType = "GRanges")
    colnames(mcols(bac)) = paste0("counts.bg.", colnames(mcols(bac)))
```

```
    mcols(bac)$geneID = sapply(strsplit(names(bac),"\\."), `[`, 1)
    # mcols(bac)$geneID = names(bac)
    bacCounts = as.data.frame(mcols(bac))
    bacCounts = bacCounts %>%
        group_by(geneID) %>%
        summarize_if(is.numeric, sum) %>%
        as.data.frame()

    # count crosslinks in binding sites
    bs = unlist(bs)
    bsInitial = bs
    bsCounts = coverageBySignal(range = bs, signal = signal, returnType = "GRanges")
    colnames(mcols(bsCounts)) = paste0("counts.bs.", colnames(mcols(bsCounts)))

    # set matching IDs
    bsCounts$geneID = sapply(strsplit(names(bsCounts),"\\."), `[`, 1)
    bsCounts$PeakID = names(bsCounts)
    bsCounts = as.data.frame(mcols(bsCounts))

    # match peak and gene counts
    idx = match(bsCounts$geneID, bacCounts$geneID)
    comb = cbind.data.frame(bsCounts, bacCounts[idx,])
    comb$geneID = NULL
    comb$PeakID = NULL

    # combine counts and peak ranges for output
    idx = match(names(bs), rownames(comb))
    mcols(bs) = comb[idx,]

    # remove duplicated binding sites
    if (any(duplicated(bs))) {
        message(paste0("Found ", length(duplicated(bs)[duplicated(bs) == TRUE]),
                       " duplicated ranges. These are removed. "))
        bsDub = bs[duplicated(bs)]
        bs = bs[! bs %in% bsDub]
    }


    # match binding site counts and existing meta data
    idx = match(names(bs), names(bsInitial))
    mcols(bs) = cbind(mcols(bsInitial[idx]), mcols(bs))

    return(bs)
}


countObj = makeBsBackgroundMatrix(object = bdsFinal, geneRanges = gns, offset = 5, matchBy = "geneID")
```

4.2 Differential analysis of binding sites

```
### ============================================================================
### Run test with DESeq
### ----------------------------------------------------------------------------
```

```
#
count_matrix = as.data.frame(mcols(countObj))
count_matrix = count_matrix %>% select(starts_with("counts"))

# internal modification for col data
colDataMod = rbind.data.frame(colData,colData)
colDataMod$type = c(rep("bs", nrow(colData)),
                    rep("bg", nrow(colData)))
colDataMod$condition = factor(colDataMod$condition, levels = c("KO", "WT"))
colDataMod$type = factor(colDataMod$type, levels = c("bs", "bg"))

# create SE object
se = SummarizedExperiment(assays = list(counts = as.matrix(count_matrix)),
                          rowRanges = granges(countObj), colData = colDataMod)

# set design - YOUs design
ddsBs = DESeqDataSet(se, design = ~condition + type + condition:type)
ddsBs$condition = relevel(ddsBs$condition, "WT")
ddsBs$type = relevel(ddsBs$type, "bg")
ddsBs = DESeq(ddsBs, test="LRT", reduced = ~ condition + type)
resBs = results(ddsBs, name = "conditionKO.typebs") # needs relevel of type to bg
resShrinkBs = lfcShrink(ddsBs, res = resBs, coef = "conditionKO.typebs", type = "ashr")


# deseq fetch results
diff_bs = countObj
colnames(resBs) = paste0("resBs.", colnames(resBs))
idx = match(names(diff_bs), rownames(resBs))
mcols(diff_bs) = cbind(mcols(diff_bs), resBs[idx,])
```

4.3 Differential analysis of background

```
### ============================================================================
### Run Background test with DESeq
### ----------------------------------------------------------------------------
#
# construct background signal dataframe to test for the background level change
count_matrix_bg = as.data.frame(mcols(countObj)) %>%
    select(starts_with("counts.bg")) %>%
    unique()
#rownames(count_matrix_bg) = sapply(strsplit(rownames(count_matrix_bg),"\\."), `[`, 1)

# create SE object
seBg = SummarizedExperiment(assays = list(counts = as.matrix(count_matrix_bg)), colData = colData)

# DESeq design
ddsBg = DESeqDataSet(seBg, design = ~ condition)
ddsBg$condition = relevel(ddsBg$condition, "WT")
ddsBg = DESeq(ddsBg)
resBg = results(ddsBg, contrast = c("condition", "KO", "WT"))
resShrinkBg = lfcShrink(ddsBg, res = resBg, contrast = c("condition", "KO", "WT"), type = "ashr")

# add results to final differential object
```

```
colnames(resBg) = paste0("resBg.", colnames(resBg))
idx = match(names(diff_bs), rownames(resBg))
mcols(diff_bs) = cbind(mcols(diff_bs), resBg[idx,])
```

```
# ### ======================================================================
# ### Run enhanced background test with DESeq
# ### ----------------------------------------------------------------------
# #
# # load WT
# peaksInitialWT = "/Users/mirko/Projects/mirEClip/01_eCLIP/pureclip/IP_WT_pureclip_sites.bed"
# peaksInitialWT = import(con = peaksInitialWT, format = "BED", extraCols=c("additionalScores" = "chara
# peaksInitialWT = keepStandardChromosomes(peaksInitialWT, pruning.mode = "coarse")
# # load KO
# peaksInitialKO = "/Users/mirko/Projects/mirEClip/01_eCLIP/pureclip/IP_KO_pureclip_sites.bed"
# peaksInitialKO = import(con = peaksInitialKO, format = "BED", extraCols=c("additionalScores" = "chara
# peaksInitialKO = keepStandardChromosomes(peaksInitialKO, pruning.mode = "coarse")
# # peaks all
# peaksAll = unique(c(peaksInitialKO, peaksInitialWT))
# bdsPeaksAll = setRanges(bds, peaksAll)
#
# countObj2 = makeBsBackgroundMatrix(object = bdsPeaksAll, geneRanges = gns, offset = 5, matchBy = "ran
# # construct background signal dataframe to test for the background level change
# mAllPeaks = as.data.frame(mcols(countObj2))
# mAllPeaks = mAllPeaks %>%
#     select(starts_with("counts.bg")) %>%
#     unique()
# rownames(mAllPeaks) = sapply(strsplit(rownames(mAllPeaks),"\\."), `[`, 1)
#
# # create SE object
# seBg2 = SummarizedExperiment(assays = list(counts = as.matrix(mAllPeaks)), colData = colData)
#
# # DESeq design
# ddsBg2 = DESeqDataSet(seBg2, design = ~ condition)
# ddsBg2$condition = relevel(ddsBg2$condition, "WT")
# ddsBg2 = DESeq(ddsBg2)
# resBg2 = results(ddsBg2, contrast = c("condition", "KO", "WT"))
# resBg2 = lfcShrink(ddsBg2, res = resBg2, contrast = c("condition", "KO", "WT"), type = "ashr")


### ======================================================================
### Run Gene test with DESeq
### ----------------------------------------------------------------------
#
# # construct gene signal dataframe to test for total gene level change
# m = mcols(countObj) %>%
#     as.data.frame() %>%
#     select(starts_with("counts.bs"), geneID) %>%
#     group_by(geneID) %>%
#     summarise_all(sum) %>%
#     tibble::column_to_rownames("geneID")
#
# # create SE object
# seGene = SummarizedExperiment(assays = list(counts = as.matrix(m)), colData = colData)
#
```

Table 2: Results by numbers. Significant based on IHW adjusted P value threshold 0.05.

Name	Count (#N)	Percentage (%)
Tested	28,253	100.00
Not Significant	27,115	95.97
Significant	1,138	4.03
Sig+Up	296	26.01
Sig+Down	842	73.99

```
# # DESeq design
# ddsGene = DESeqDataSet(seGene, design = ~ condition)
# ddsGene$condition = relevel(ddsGene$condition, "WT")
# ddsGene = DESeq(ddsGene)
# resGene = results(ddsGene, contrast = c("condition", "KO", "WT"))
# resGene = lfcShrink(ddsGene, res = resGene, contrast = c("condition", "KO", "WT"), type = "ashr")


### ============================================================================
### Export results
### ----------------------------------------------------------------------------
#
```

4.4 Initial results

```
### ============================================================================
### Numbers overview table
### ----------------------------------------------------------------------------
###
df = data.frame(Name = c("Tested", "Not Significant", "Significant", "Sig+Up", "Sig+Down"),
                N = c(nrow(resBs),
                      nrow(subset(resBs, resBs.padj >= 0.05)),
                      nrow(subset(resBs, resBs.padj < 0.05)),
                      nrow(subset(resBs, resBs.padj < 0.05 & resBs.log2FoldChange > 0)),
                      nrow(subset(resBs, resBs.padj < 0.05 & resBs.log2FoldChange < 0))
                      ))
df$Per = c(100,
           nrow(subset(resBs, resBs.padj >= 0.05)) / nrow(resBs) * 100,
           nrow(subset(resBs, resBs.padj < 0.05)) / nrow(resBs) * 100,
           nrow(subset(resBs, resBs.padj < 0.05 & resBs.log2FoldChange > 0)) / nrow(subset(resBs, resBs
           nrow(subset(resBs, resBs.padj < 0.05 & resBs.log2FoldChange < 0)) / nrow(subset(resBs, resBs
           )
df$Per = round(df$Per, digits = 2)
colnames(df) = c("Name", "Count (#N)", "Percentage (%)")

kable(myNumberFormat(df), caption = "Results by numbers. Significant based on IHW adjusted P value thres
```

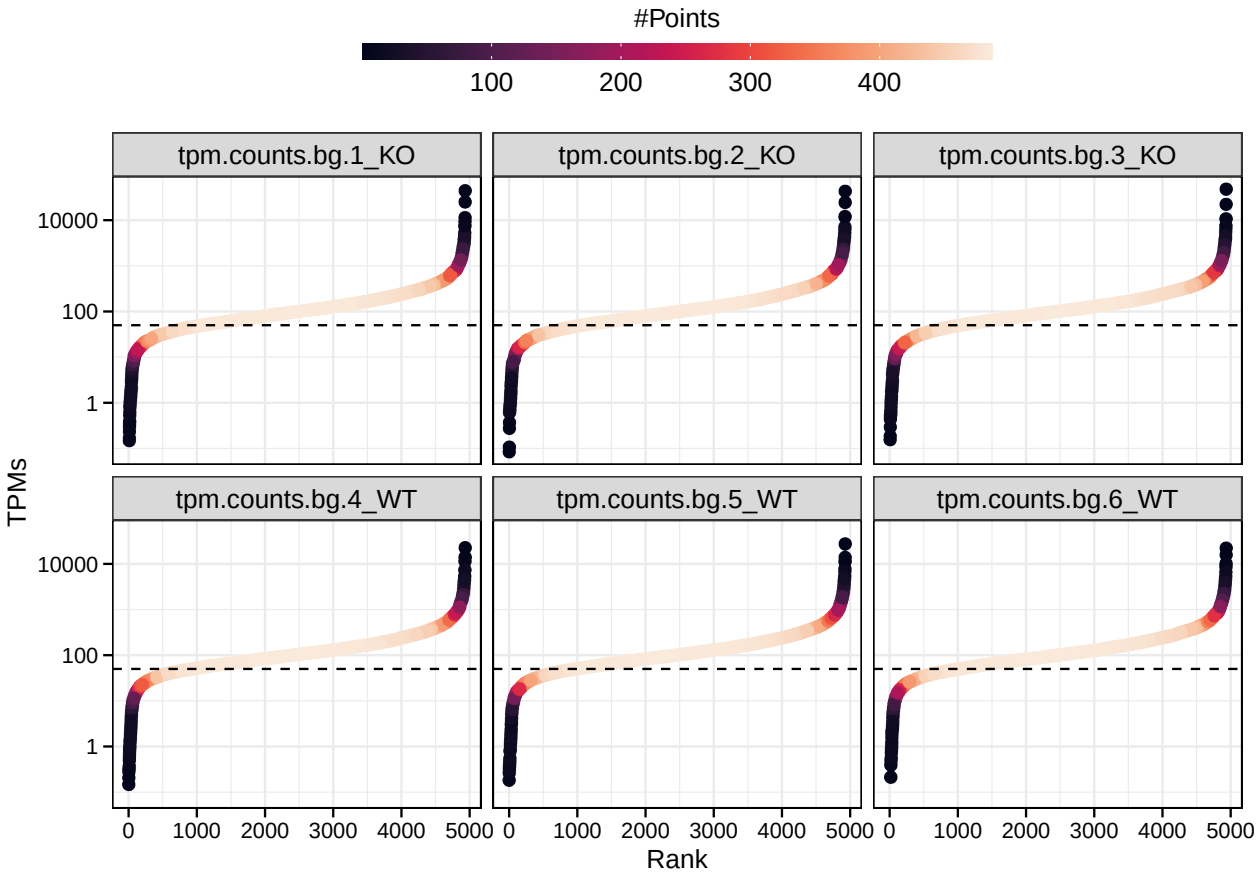
4.5 Gene expression filter

Some genes might not be expressed in both conditions. Binding sites on a gene that is for example only expressed in the *WT* but not in the *KO* condition will all appear to be down regulated. In reality we can not tell if such sites changed, because the hosting gene is not expressed. To prevent these effects I filtered all genes with a merged binding site (so the combined set from both conditions) before running the differential

binding search. As cutoff a TPM > 50 is used.

```
# cutoff based on gene

counts = counts(ddsBg, normalized = FALSE)
# Extract all exons of a gen
exonsByGene <- exonsBy(annoDb, by = "gene")
# reduce overlapping exons to a single region
reducedExonsByGene <- reduce(exonsByGene)
# Approximate the gene length as the sum of the its exons lengths
geneLengths <- sum(width(reducedExonsByGene))
# Adapt gene IDs
names(geneLengths) = sapply(strsplit(names(geneLengths),"\\."), `[`, 1)
# Re-order the vector of gene lengths to match the order in the counts
counts_genes <- sapply(strsplit(rownames(counts),"\\."),  `[`, 1)
geneLengths <- geneLengths[match(counts_genes, names(geneLengths))]
# First step1 in TPM calculation
tpms <- counts / geneLengths
# Second step in TPM calculation
tpms <- t(t(tpms) * 1e6 / colSums(tpms, na.rm = T)) %>% as.data.frame()
# # Adapt gene IDs
rownames(tpms) = sapply(strsplit(rownames(tpms),"\\."), `[`, 1)
colnames(tpms) = paste0("tpm.", colnames(tpms) )

# add tpm to final object
idx <- match(diff_bs$geneID, rownames(tpms))
mcols(diff_bs) <- cbind(mcols(diff_bs), tpms[idx,])

df = tpms %>%
    as.data.frame() %>%
    tibble::rownames_to_column("geneID") %>%
    pivot_longer(-geneID) %>%
    group_by(name) %>%
    mutate(rank = rank(value, ties.method = "first"))


ggplot(df, aes(x = rank, y = value)) +
    ggrastr::rasterise(geom_pointdensity(size = 2), dpi = 300) +
    facet_wrap(~name) +
    theme_nice() +
    geom_hline(yintercept = 50, linetype = "dashed") +
    scale_y_log10() +
    scale_color_viridis(option = "rocket") +
    theme(legend.position = "top") +
    guides(color = guide_colorbar(title.position = 'top', title.hjust = 0.5,
                                  barwidth = unit(20, 'lines'), barheight = unit(.5, 'lines'))) +
    labs(
        title = "TPMs per gene per sample",
        x = "Rank",
        y = "TPMs",
        color = "#Points"
        )
```

TPMs per gene per sample



A gene is considered expressed by a condition if at least two of the three replicates have a TPM over 50. Each gene must be found expressed in both conditions to be considered for the differential binding analysis.

```
# filter for tpm cutoff in 2 samples per condition
diff_bs$BS_ID <- names(diff_bs)
diff_bs <- as.data.frame(diff_bs) %>%
  rowwise() %>%
  mutate(tpm_support_KO = sum(c((tpm.counts.bg.1_KO > tpm_cut),
                                (tpm.counts.bg.2_KO > tpm_cut),
                                (tpm.counts.bg.3_KO > tpm_cut))),
         tpm_support_WT = sum(c((tpm.counts.bg.4_WT > tpm_cut),
                                (tpm.counts.bg.5_WT > tpm_cut),
                                (tpm.counts.bg.6_WT > tpm_cut))),
         tpm_supported = (tpm_support_KO > 1) & (tpm_support_WT > 1))


# make upset plot
genesExpInWT = diff_bs %>% subset(tpm_support_WT > 1) %>% pull(geneID) %>% unique()
genesExpInKO = diff_bs %>% subset(tpm_support_KO > 1) %>% pull(geneID) %>% unique()

l = list(genesExpInWT = genesExpInWT, genesExpInKO = genesExpInKO)
m = make_comb_mat(l)

ha = HeatmapAnnotation(
  "Intersections" = anno_barplot(comb_size(m), border = FALSE, gp = gpar(fill = "#a6a6a6"), height = un
```

```
)
ha2 = HeatmapAnnotation(
  "set sizes" = anno_barplot(set_size(m), border = FALSE, gp = gpar(fill = "#a6a6a6"), width = unit(4,
  which = "row"
)

ht = UpSet(m,
           comb_order = order(comb_size(m), decreasing = T),
           top_annotation = ha,
           right_annotation = ha2,
           comb_col = "#003399", bg_col = "white", pt_size = unit(.5, "cm") ,
           border = T, lwd = 2, bg_pt_col = "#a6a6a6"
)

ss = set_size(m)
cs = comb_size(m)
ht = draw(ht,  padding = unit(c(0, 0, 10, 0), "mm"))
od = column_order(ht)
decorate_annotation("Intersections", {
    grid.text(format(cs[od], big.mark = ".", decimal.mark = ","), x = seq_along(cs), y = unit(cs[od],
        default.units = "native", just = c("left", "bottom"),
        gp = gpar(fontsize = 8, col = "black"), rot = 45)
})
```

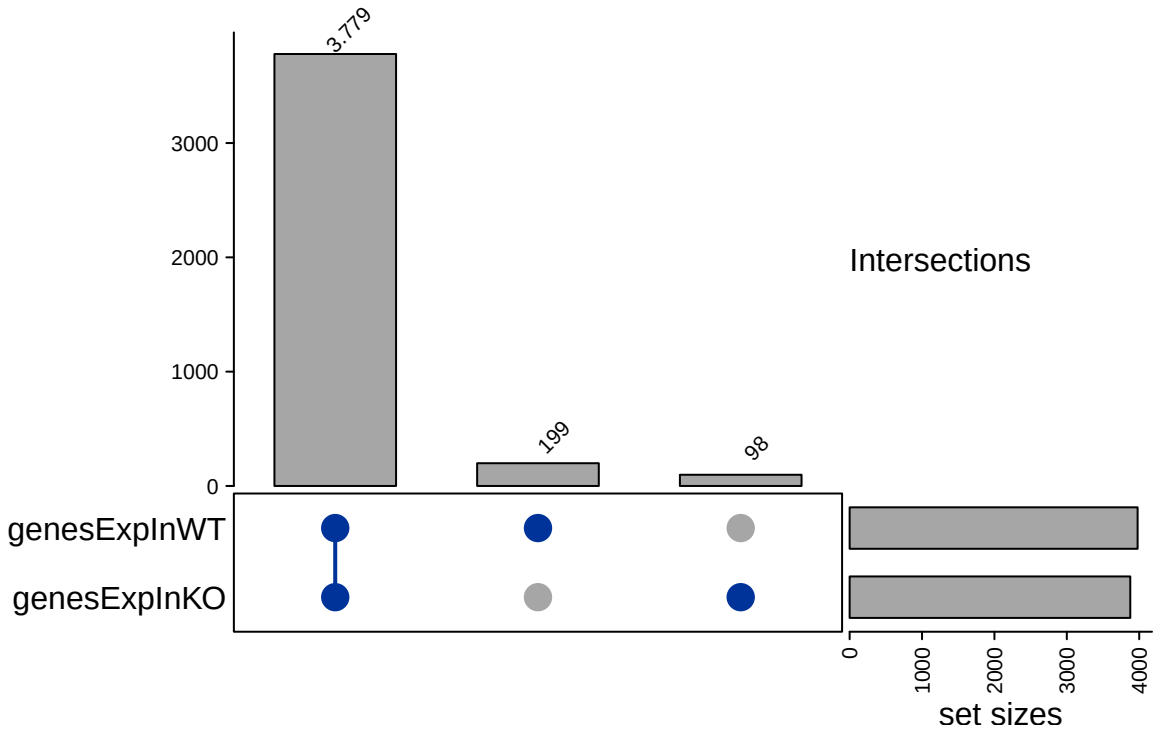


Figure 1: Intersection of genes expressed in both conditions.

```
# subset by cutoff
diff_bs <- diff_bs %>% subset(tpm_supported )
```

The expression filtering resulted in 3,779 target genes, with 26,462 binding sites.

Table 3: Results by numbers. Significant based on IHW adjusted P value threshold 0.05.

Name	Count (#N)	Percentage (%)
Tested	26,462	100.00
Not Significant	25,388	95.94
Significant	1,074	4.06
Sig+Up	278	25.88
Sig+Down	796	74.12

5 Results (after Gene Expression filter)

```
### ============================================================================
### Numbers overview table
### ----------------------------------------------------------------------------
###
df = data.frame(Name = c("Tested", "Not Significant", "Significant", "Sig+Up", "Sig+Down"),
                N = c(nrow(diff_bs),
                      nrow(subset(diff_bs, resBs.padj >= 0.05)),
                      nrow(subset(diff_bs, resBs.padj < 0.05)),
                      nrow(subset(diff_bs, resBs.padj < 0.05 & resBs.log2FoldChange > 0)),
                      nrow(subset(diff_bs, resBs.padj < 0.05 & resBs.log2FoldChange < 0))
                      ))
df$Per = c(100,
           nrow(subset(diff_bs, resBs.padj >= 0.05)) / nrow(diff_bs) * 100,
           nrow(subset(diff_bs, resBs.padj < 0.05)) / nrow(diff_bs) * 100,
           nrow(subset(diff_bs, resBs.padj < 0.05 & resBs.log2FoldChange > 0)) / nrow(subset(diff_bs, re
           nrow(subset(diff_bs, resBs.padj < 0.05 & resBs.log2FoldChange < 0)) / nrow(subset(diff_bs, re
           )
df$Per = round(df$Per, digits = 2)
colnames(df) = c("Name", "Count (#N)", "Percentage (%)")

kable(myNumberFormat(df), caption = "Results by numbers. Significant based on IHW adjusted P value thres
```

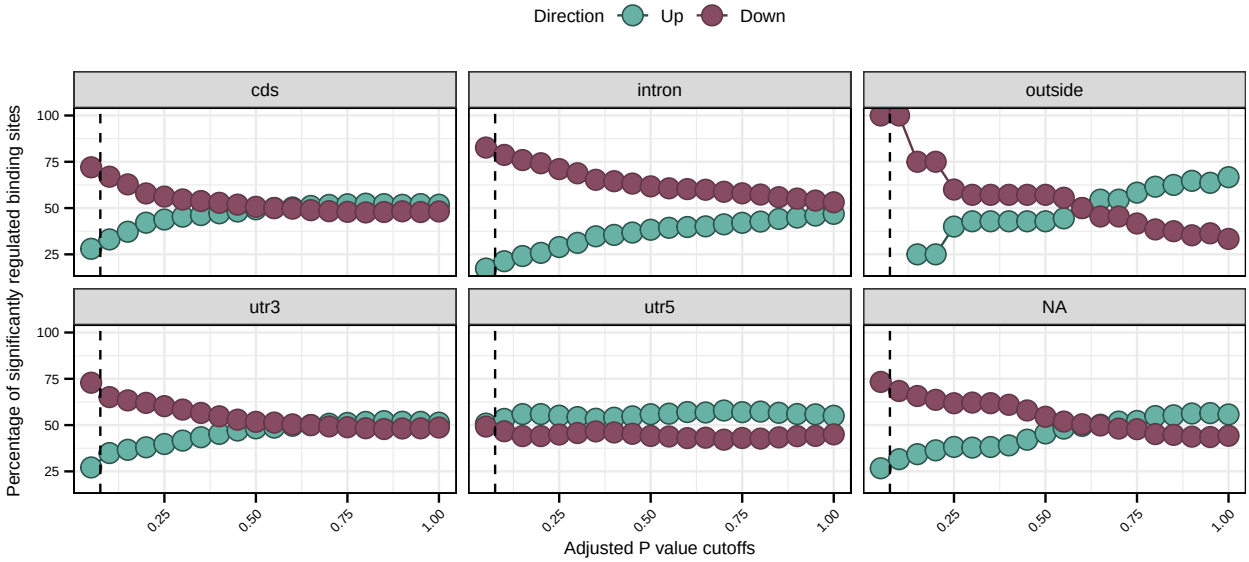
5.1 Decide for P value cutoff

```
d = lapply(seq(0, 1, by = 0.05), function(cutoff){
    diff_bs %>%
        select(region, resBs.log2FoldChange, resBs.padj) %>%
        rename("lfc" = "resBs.log2FoldChange", "padj" = "resBs.padj") %>%
        subset(padj <= cutoff) %>%
        mutate(dir = ifelse(lfc > 0, "Up", "Down")) %>%
        select(region, dir) %>%
        mutate(cutoff = cutoff)
})
d = do.call("rbind", d)

df = d %>%
    group_by(region, dir, cutoff) %>%
    tally() %>%
    ungroup() %>%
    group_by(cutoff, region) %>%
    mutate(sum = sum(n)) %>%
```

```
    arrange(cutoff) %>%
    mutate(percentage = round(n/sum, digits = 4)*100) %>%
    mutate(dir = factor(dir, level = c("Up", "Down"))) %>%
    arrange(dir)

p1 = ggplot(df, aes(x = cutoff, y = percentage, color = dir, fill = dir, group = dir)) +
    geom_line() +
    geom_point(size = 4, shape = 21) +
    theme_pub() +
    scale_fill_nice_ud_b() +
    scale_color_nice_ud_d() +
    theme(legend.position = "top") +
    labs(
        title = "Regulation enrichment",
        x = "Adjusted P value cutoffs",
        y = "Percentage of significantly regulated binding sites",
        color = "Direction",
        fill = "Direction"
    ) +
    theme(axis.text.x=element_text(angle = 45, hjust = 1, vjust = 1)) +
    facet_wrap(~region) +
    geom_vline(xintercept = 0.075, linetype = "dashed")

p2 = ggplot(df, aes(x = cutoff, y = n, fill = dir, color = dir, group = dir)) +
    geom_col() +
    theme_pub() +
    scale_fill_nice_ud_b() +
    scale_color_nice_ud_d() +
    theme(legend.position = "top") +
    theme(axis.text.x=element_text(angle = 45, hjust = 1, vjust = 1)) +
    facet_wrap(~region, scales = "free_y") +
    geom_vline(xintercept = 0.075, linetype = "dashed") +
    labs(
        title = "Number of binding site per adjusted P value cutoff",
        x = "Adjusted P value cutoffs",
        y = "Count",
        color = "Direction",
        fill = "Direction"
    )

p1 / p2
```

```
# P value hist
p1 = ggplot(diff_bs, aes(x = resBs.pvalue)) +
    geom_histogram(bins = 50, fill = "#999999", color = "#4d4d4d") +
    theme_nice() +
    labs(
        x = "P value",
        y = "Count"
    )

p2 = ggplot(diff_bs, aes(x = resBs.padj)) +
    geom_histogram(bins = 50, fill = "#999999", color = "#4d4d4d") +
    theme_nice() +
```

Regulation enrichment



Number of binding site per adjusted P value cutoff

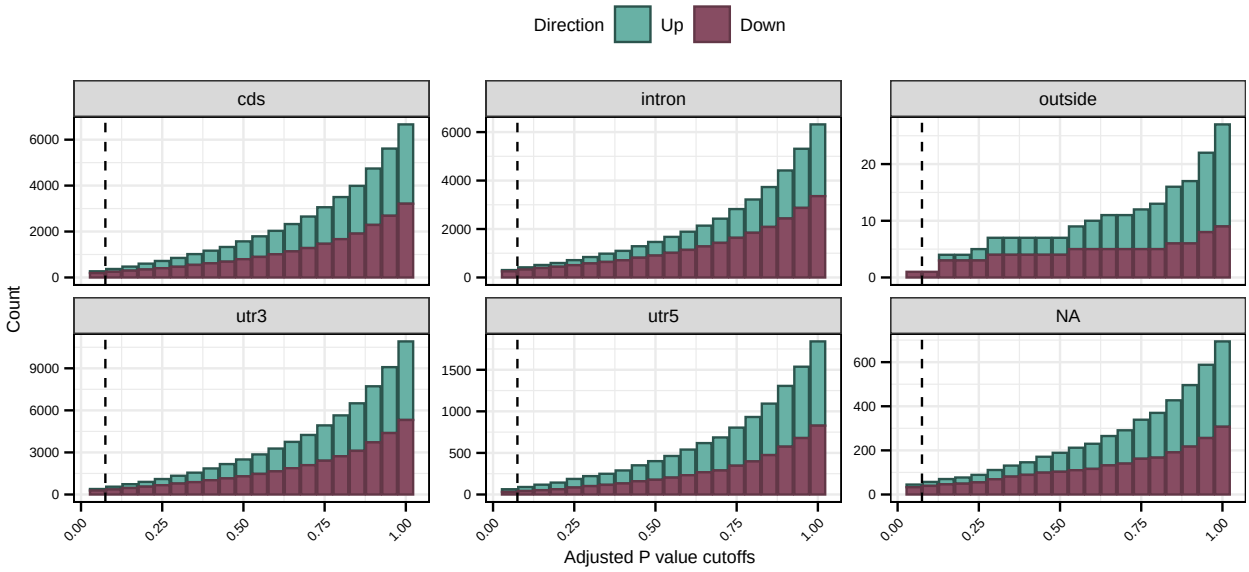


Figure 2: Regulation enrichment per adjusted P value cutoff. The dashed line indicates a cutoff of 0.05.

```
    geom_vline(xintercept = 0.05) +
    labs(
        x = "Adj P value",
        y = "Count"
    )

p1 + p2
```

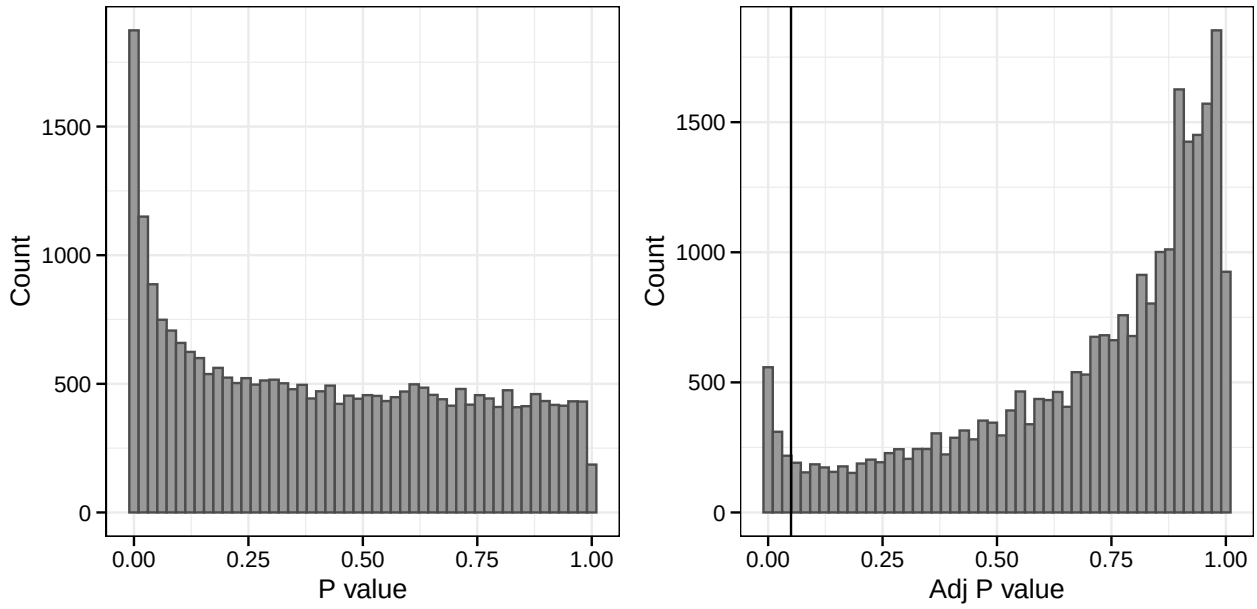


Figure 3: P value distributions.

```
df2 = d %>%
    subset(., cutoff == 0.05) %>%
    subset(., !is.na(region)) %>%
    subset(., region != "outside")%>%
    group_by(region, dir) %>%
    tally() %>%
    ungroup() %>%
    mutate(sum = sum(n)) %>%
    mutate(percentage = round(n/sum, digits = 4)*100)
df2 = df2 %>%
    mutate(region = factor(region, levels = c("utr5", "intron", "cds", "utr3"))) %>%
    arrange(region)

p1 = ggplot(df2, aes(x = dir, y = percentage, fill = region)) +
    geom_col(position = "stack") +
    theme_pub() +
    scale_fill_npg() +
    labs(
        title = "Significantly regulated binding sites per region and direction",
        x = "Direction of regulation",
        y = "Percentage",
        fill = "Region"
    ) +
```

```
    coord_flip() +
    theme(legend.position = "top")

p2 = ggplot(df2, aes(x = dir, y = percentage, fill = region)) +
    geom_col(position = "fill", lwd = 2) +
    theme_pub() +
    scale_fill_npg() +
    labs(
        title = "Significantly regulated binding sites per region and direction",
        x = "Direction of regulation",
        y = "Percentage",
        fill = "Region"
    ) +
    coord_flip() +
    theme(legend.position = "top")
    # annotate(geom = "rect", xmin = 0.55, xmax = 1.45, ymin = 0, ymax = 1, fill = "NA", color = "#874C
    # annotate(geom = "rect", xmin = 1.55, xmax = 2.45, ymin = 0, ymax = 1, fill = "NA", color = "#68b1

p1 + p2
```

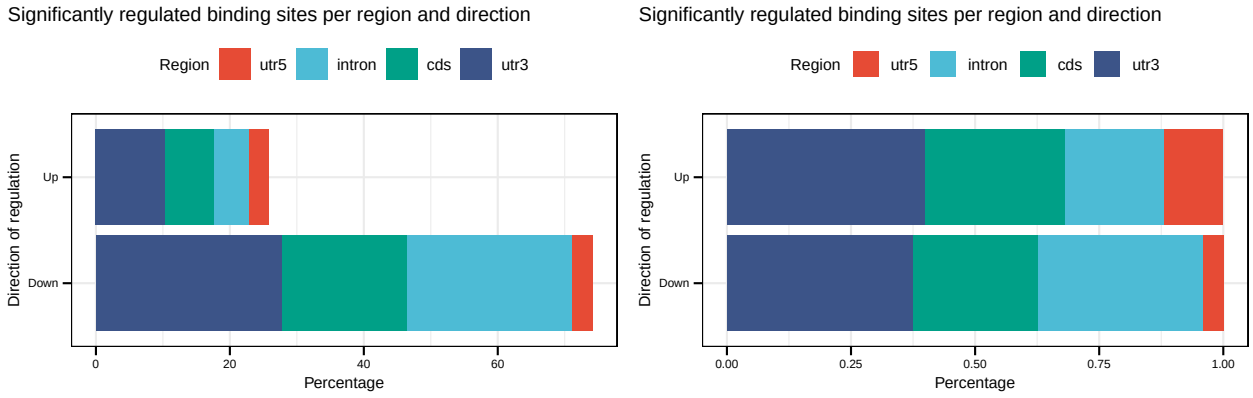


Figure 4: Significantly regulated binding sites per region and direction. The percentage is calculated from all significant binding sites at adusted P value cutoff of 0.05.

5.1.1 Decide for fold-change cutoff

```
d = lapply(log2(seq(1, 10, by = 0.5)), function(cutoff){
    diff_bs %>%
        filter(resBs.padj < 0.05 & abs(resBs.log2FoldChange) > cutoff) %>%
        mutate(dir = factor(ifelse(resBs.log2FoldChange > 0, "Up", "Down"), level = c("Up", "Down"))) %
        select(dir) %>%
        group_by(dir) %>%
        tally() %>%
        # mutate(cutoff = paste0("log2(", 2^cutoff, ")")) %>%
        mutate(cutoff = 2^cutoff ) %>%
        # mutate(cutoff = factor(cutoff, levels = c(paste0("log2(", seq(1, 10, by = 0.5), ")")))) %>%
        mutate(cutoff = factor(cutoff, levels = c(paste0(seq(1, 10, by = 0.5))))) %>%
        arrange(cutoff)
})
d = do.call("rbind", d)
```

```r
df = d %>%
    group_by(cutoff) %>%
    mutate(sum = sum(n)) %>%
    arrange(cutoff) %>%
    mutate(percentage = round(n/sum, digits = 4)*100)

p1 = ggplot(df, aes(x = cutoff, y = percentage, color = dir, fill = dir, group = dir)) +
    geom_line() +
    geom_point(size = 4, shape = 21) +
    theme_pub() +
    scale_fill_nice_ud_b() +
    scale_color_nice_ud_d() +
    theme(legend.position = "top") +
    # theme(axis.text.x=element_text(angle = 45, hjust = 1, vjust = 1)) +
    labs(
        title = "Regulation enrichment",
        x = "Fold-change cutoffs (log2)",
        y = "Percentage of regulated binding sites",
        color = "Direction",
        fill = "Direction"
    )

p2 = ggplot(df, aes(x = cutoff, y = n, fill = dir, color = dir, group = dir)) +
    geom_col() +
    theme_pub() +
    scale_fill_nice_ud_b() +
    scale_color_nice_ud_d() +
    theme(legend.position = "top") +
    # theme(axis.text.x=element_text(angle = 45, hjust = 1, vjust = 1)) +
    labs(
        title = "Number of binding site per fold-change cutoff",
        x = "Fold-change cutoffs (log2)",
        y = "Count",
        color = "Direction",
        fill = "Direction"
    )

p1 + p2
```

```r
df = diff_bs %>%
    mutate(dir = factor(ifelse(resBs.log2FoldChange > 0, "Up", "Down"), level = c("Up", "Down"))) %>%
    arrange(dir)
p1 = ggplot(df, aes(x = resBs.log2FoldChange)) +
    geom_histogram(bins = 50, fill = "#999999", color = "#4d4d4d") +
    theme_nice() +
    labs(
        x = "Fold-change (log2)",
        y = "Count"
    )

df = diff_bs %>%
    mutate(sig = factor(ifelse(resBs.padj < 0.05, TRUE, FALSE))) %>%
    mutate(dir = factor(ifelse(resBs.log2FoldChange > 0, "Up", "Down"), level = c("Up", "Down"))) %>%
    arrange(dir)
```

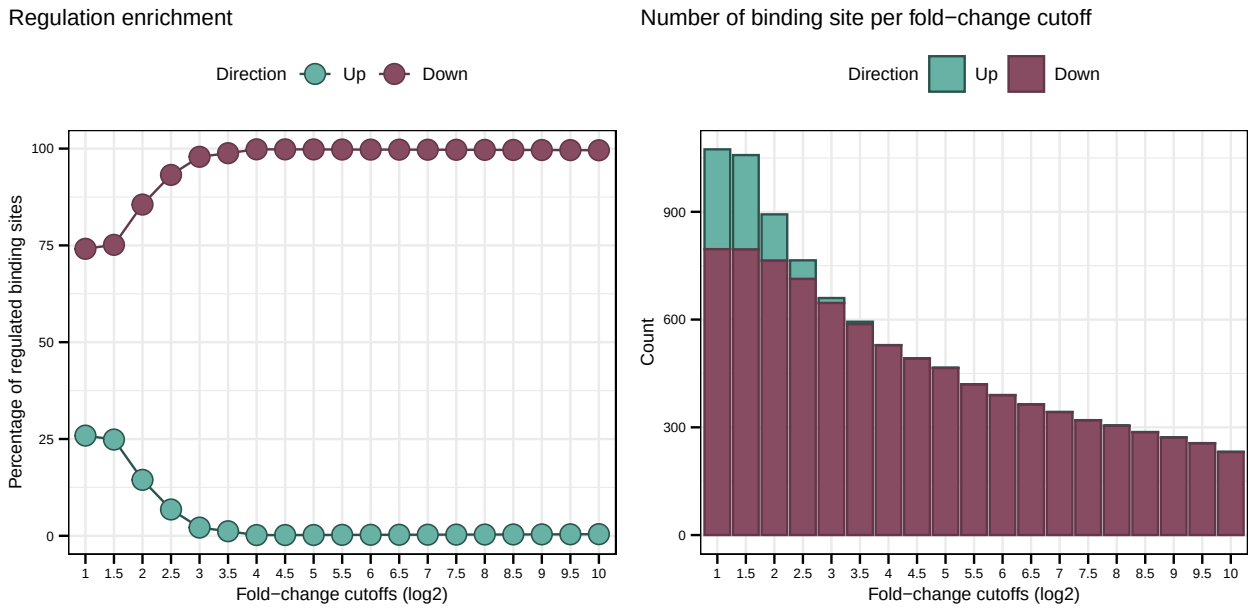


Figure 5: Enrichment of the direction of regulation. Based on differnt cutoffs for the fold-change per binding site.

```
p2 = ggplot(subset(df, sig == TRUE), aes(x = resBs.log2FoldChange, fill = dir, color = dir)) +
    geom_histogram(bins = 100) +
    scale_fill_nice_ud_b() +
    scale_color_nice_ud_d() +
    theme_nice() +
    theme(legend.position = "top") +
    labs(
        x = "Fold-change (log2)",
        y = "Count",
        color = "Regulation",
        fill = "Regulation"
    )

df = diff_bs %>%
    mutate(sig = factor(ifelse(resBs.padj < 0.05 & abs(resBs.log2FoldChange) > log2(2), TRUE, FALSE))) %
    mutate(dir = factor(ifelse(resBs.log2FoldChange > 0, "Up", "Down"), level = c("Up", "Down"))) %>%
    arrange(dir)
p3 = ggplot(subset(df, sig == TRUE), aes(x = resBs.log2FoldChange, fill = dir, color = dir)) +
    geom_histogram(bins = 100) +
    scale_fill_nice_ud_b() +
    scale_color_nice_ud_d() +
    theme_nice() +
    theme(legend.position = "top") +
    labs(
        x = "Fold-change (log2)",
        y = "Count",
        color = "Regulation",
        fill = "Regulation"
    )
p1 + p2 + p3
```

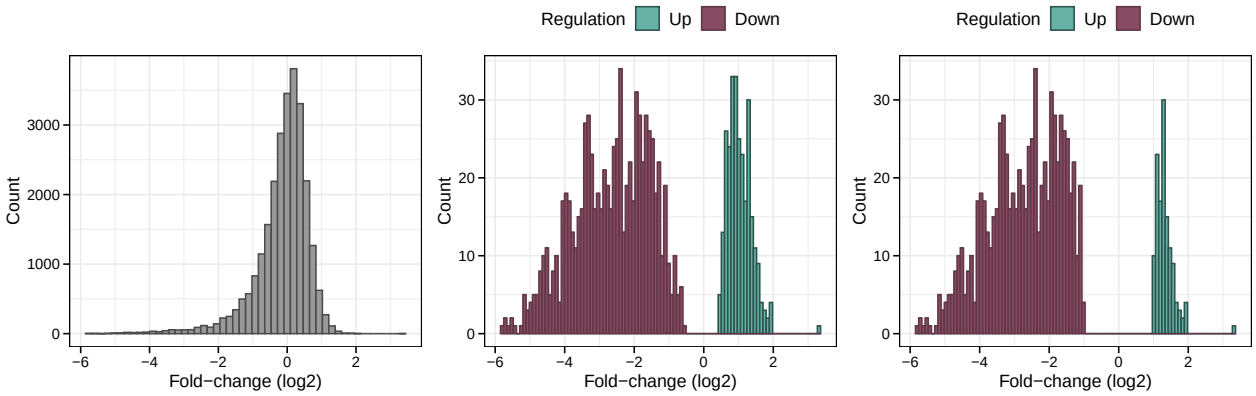


Figure 6: Fold-change distributions. Fold-Change cutoff in the last plot = 2).

5.1.2 Final results plots

```
# MA + Volcano plots
df = diff_bs %>%
    mutate(sig = (ifelse(resBs.padj < 0.05 & abs(resBs.log2FoldChange) > log2(2), TRUE, FALSE))) %>%
    mutate(dir = factor(ifelse(resBs.log2FoldChange > 0, "Up", "Down"), level = c("Up", "Down"))) %>%
    mutate(sigDir = ifelse(sig == TRUE & dir == "Up", "Up", ifelse(sig == TRUE & dir == "Down", "Down",
    mutate(sigDir = factor(sigDir, levels = c("Not", "Up", "Down"))) %>%
    arrange(sigDir)
p1 = ggplot(df, aes(x = log2(resBs.baseMean), y = resBs.log2FoldChange, color = sigDir, fill = sigDir))
  geom_point_rast(shape = 21, stroke = 0.5, size = 1.5) +
  scale_fill_nice_udn_b() +
  scale_color_nice_udn_d() +
  geom_hline(yintercept = 0, color = "black", alpha = .5) +
  theme_nice() +
  guides(color = guide_legend(override.aes = list(size = 4))) +
  theme(legend.key.size = unit(1, 'cm'), legend.position = "top") +
  labs(
      x = "Base mean",
      y = "Fold-change (log2)",
      color = "Regulation",
      fill = "Regulation")

p2 = ggplot(df, aes(x = resBs.log2FoldChange, y = -log10(resBs.padj), color = sigDir, fill = sigDir)) +
    geom_point_rast(shape = 21, stroke = 0.5, size = 1.5) +
    scale_fill_nice_udn_b() +
    scale_color_nice_udn_d() +
    geom_vline(xintercept = 0, color = "black", alpha = .5) +
    theme_nice() +
    guides(color = guide_legend(override.aes = list(size = 4))) +
    theme(legend.key.size = unit(1, 'cm'), legend.position = "top") +
    labs(
        x = "Fold-change (log2)",
        y = "Adjusted P value (-log10)",
        color = "Regulation",
        fill = "Regulation")

(p1 + p2) + plot_layout(guides = "collect") & theme(legend.position = "top")
```

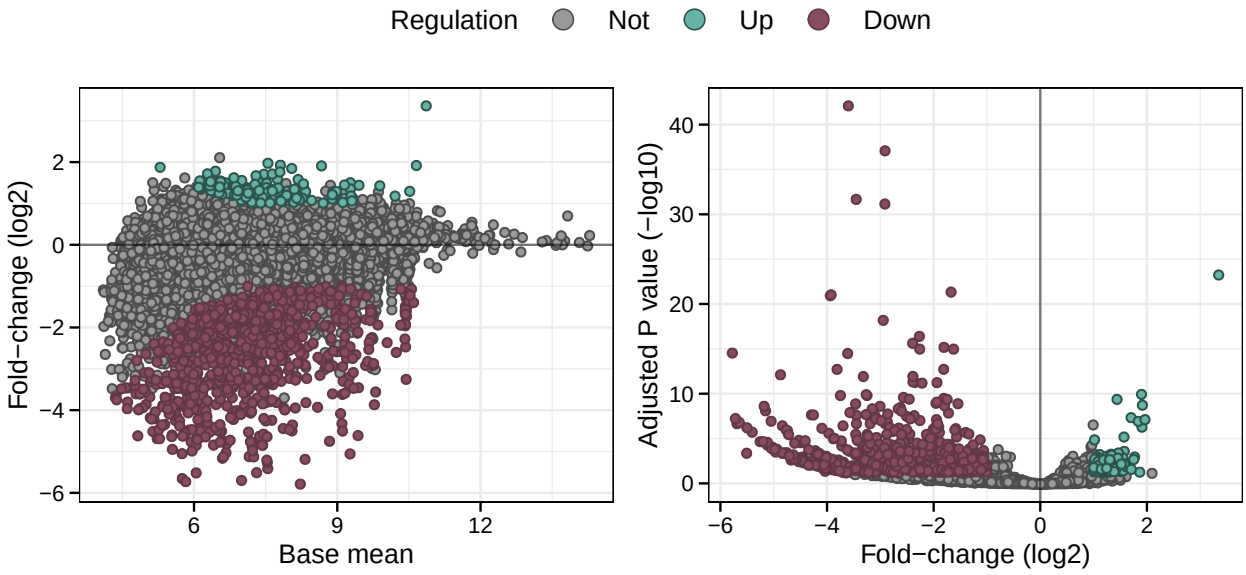


Figure 7: MA and Volcano plots. With adjusted P value and fold-change cutoffs. Simple version.

```
p2 <- p2+theme_paper()
ggsave(p2, file= paste0(out, "Figure1X_Differntial_binding_vulcano.pdf"), height = 6, width = 6, units =
```

```
# MA + Volcano plots
df = diff_bs %>%
    mutate(sig = (ifelse(resBs.padj < 0.05 & abs(resBs.log2FoldChange) > log2(2), TRUE, FALSE))) %>%
    mutate(dir = factor(ifelse(resBs.log2FoldChange > 0, "Up", "Down"), level = c("Up", "Down"))) %>%
    mutate(sigDir = ifelse(sig == TRUE & dir == "Up", "Up", ifelse(sig == TRUE & dir == "Down", "Down",
    mutate(sigDir = factor(sigDir, levels = c("Not", "Up", "Down"))) %>%
    arrange(sigDir)

dfN = df %>% subset(sig == TRUE) %>% select(dir) %>% group_by(dir) %>% tally()

dfLabMa_up = df %>% subset(sigDir == "Up") %>%
    arrange(desc(resBs.log2FoldChange) ) %>%
    head(5)
dfLabMa_down = df %>% subset(sigDir == "Down") %>%
    arrange((resBs.log2FoldChange) ) %>%
    head(5)


p1 = ggplot(df, aes(x = log2(resBs.baseMean), y = resBs.log2FoldChange, color = sigDir, fill = sigDir))
    geom_point_rast(shape = 21, stroke = 0.5, size = 1.5) +
    scale_fill_nice_udn_b() +
    scale_color_nice_udn_d() +
    geom_hline(yintercept = 0, color = "black", alpha = .5) +
    theme_nice() +
    guides(color = guide_legend(override.aes = list(size = 4))) +
    theme(legend.key.size = unit(1, 'cm'), legend.position = "top") +
    labs(
        x = "Base mean",
        y = "Fold-change (log2)",
```

```
        color = "Regulation",
        fill = "Regulation") +
    geom_text_repel(data = dfLabMa_up, aes(x = log2(resBs.baseMean), y = resBs.log2FoldChange, color = s
                    min.segment.length = 0, nudge_y = 2, size = 2.5) +
    geom_text_repel(data = dfLabMa_down, aes(x = log2(resBs.baseMean), y = resBs.log2FoldChange, color =
                    min.segment.length = 0, nudge_y = -2, size = 2.5) +
    ylim(-8,8) +
    annotate(geom = "text", label = paste0("#N=",dfN[1,][[2]]), x = 12.5, y = 5) +
    annotate(geom = "text", label = paste0("#N=",dfN[2,][[2]]), x = 12.5, y = -5)

dfLabVol_up = df %>% filter(sigDir == "Up") %>%
    arrange((resBs.padj) ) %>%
    head(5)
dfLabVol_down = df %>% filter(sigDir == "Down") %>%
    arrange((resBs.padj) ) %>%
    head(5)

p2 = ggplot(df, aes(x = resBs.log2FoldChange, y = -log10(resBs.padj), color = sigDir, fill = sigDir)) +
    geom_point_rast(shape = 21, stroke = 0.5, size = 1.5) +
    scale_fill_nice_udn_b() +
    scale_color_nice_udn_d() +
    geom_vline(xintercept = 0, color = "black", alpha = .5) +
    theme_nice() +
    guides(color = guide_legend(override.aes = list(size = 4))) +
    theme(legend.key.size = unit(1, 'cm'), legend.position = "top") +
    labs(
        x = "Fold-change (log2)",
        y = "Adjusted P value (-log10)",
        color = "Regulation",
        fill = "Regulation") +
    geom_text_repel(data = dfLabVol_up, aes(x = resBs.log2FoldChange, y = -log10(resBs.padj), color = si
                    min.segment.length = 0, nudge_x = 2, size = 2.5) +
    geom_text_repel(data = dfLabVol_down, aes(x = resBs.log2FoldChange, y = -log10(resBs.padj), color =
                    min.segment.length = 0, nudge_x = -2, size = 2.5) +
    xlim(-6,6) +
    annotate(geom = "text", label = paste0("#N=",dfN[1,][[2]]), y = 50, x = 2.5) +
    annotate(geom = "text", label = paste0("#N=",dfN[2,][[2]]), y = 50, x = -2.5)

(p1 + p2) + plot_layout(guides = "collect") & theme(legend.position = "top")
```

```
# MA + Volcano plots
df = diff_bs %>%
    mutate(sig = (ifelse(resBs.padj < 0.05 & abs(resBs.log2FoldChange) > log2(2), TRUE, FALSE))) %>%
    mutate(dir = factor(ifelse(resBs.log2FoldChange > 0, "Up", "Down"), level = c("Up", "Down"))) %>%
    mutate(sigDir = ifelse(sig == TRUE & dir == "Up", "Up", ifelse(sig == TRUE & dir == "Down", "Down",
    mutate(sigDir = factor(sigDir, levels = c("Not", "Up", "Down"))) %>%
    arrange(sigDir)

dfN = df %>% filter(sig == TRUE) %>% select(dir) %>% group_by(dir) %>% tally()

selectedGenes = c("Zfp36l1","Zfp36l2","Dusp10","Ddit4","Gnb4","S1pr1")

dfLabMa_up = df %>% filter(geneName %in% selectedGenes & resBs.log2FoldChange > 0 & sig == TRUE)
dfLabMa_down = df %>% filter(geneName %in% selectedGenes & resBs.log2FoldChange < 0 & sig == TRUE)
```

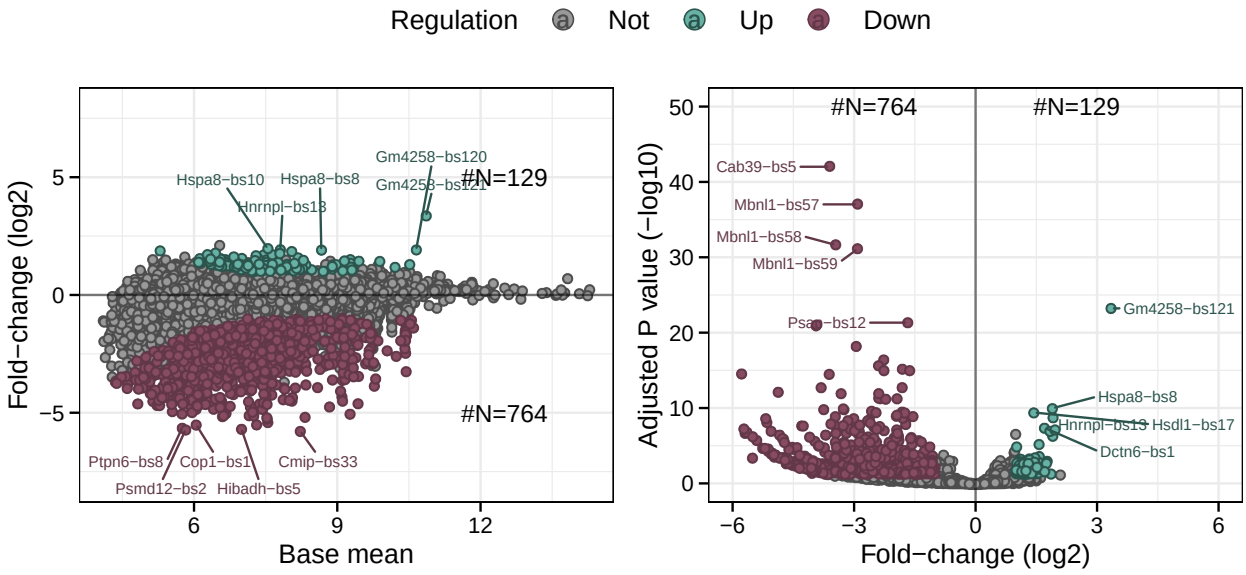


Figure 8: MA and Volcano plots. With adjusted P value and fold-change cutoffs. Additional annotaitons.

```
p1 = ggplot(df, aes(x = log2(resBs.baseMean), y = resBs.log2FoldChange, color = sigDir, fill = sigDir))
    geom_point_rast(shape = 21, stroke = 0.5, size = 1.5) +
    scale_fill_nice_udn_b() +
    scale_color_nice_udn_d() +
    geom_hline(yintercept = 0, color = "black", alpha = .5) +
    theme_nice() +
    guides(color = guide_legend(override.aes = list(size = 4))) +
    theme(legend.key.size = unit(1, 'cm'), legend.position = "top") +
    labs(
        x = "Base mean",
        y = "Fold-change (log2)",
        color = "Regulation",
        fill = "Regulation") +
    # geom_text_repel(data = dfLabMa_up, aes(x = log2(res.baseMean), y = res.log2FoldChange, color = si
    #                 min.segment.length = 0, nudge_y = 6, size = 2.5) +
    geom_text_repel(data = dfLabMa_down, aes(x = log2(resBs.baseMean), y = resBs.log2FoldChange, color =
                    min.segment.length = 0, nudge_y = -6, size = 2.5) +
    ylim(-7,7) +
    annotate(geom = "text", label = paste0("#N=",dfN[1,][[2]]), x = 12.5, y = 5) +
    annotate(geom = "text", label = paste0("#N=",dfN[2,][[2]]), x = 12.5, y = -5)


p2 = ggplot(df, aes(x = resBs.log2FoldChange, y = -log10(resBs.padj), color = sigDir, fill = sigDir)) +
    geom_point_rast(shape = 21, stroke = 0.5, size = 1.5) +
    scale_fill_nice_udn_b() +
    scale_color_nice_udn_d() +
    geom_vline(xintercept = 0, color = "black", alpha = .5) +
    theme_nice() +
    guides(color = guide_legend(override.aes = list(size = 4))) +
    theme(legend.key.size = unit(1, 'cm'), legend.position = "top") +
    labs(
```

```
        x = "Fold-change (log2)",
        y = "Adjusted P value (-log10)",
        color = "Regulation",
        fill = "Regulation") +
    # geom_text_repel(data = dfLabMa_up, aes(x = res.log2FoldChange, y = -log10(res.padj), color = sigD
    #                 min.segment.length = 0, nudge_x = 2, size = 2.5) +
    geom_text_repel(data = dfLabMa_down, aes(x = resBs.log2FoldChange, y = -log10(resBs.padj), color = s
                    min.segment.length = 0, nudge_x = -4, size = 2.5) +
    xlim(-6,6) +
    annotate(geom = "text", label = paste0("#N=",dfN[1,][[2]]), y = 50, x = 2.5) +
    annotate(geom = "text", label = paste0("#N=",dfN[2,][[2]]), y = 50, x = -2.5)

(p1 + p2) + plot_layout(guides = "collect") & theme(legend.position = "top")
```

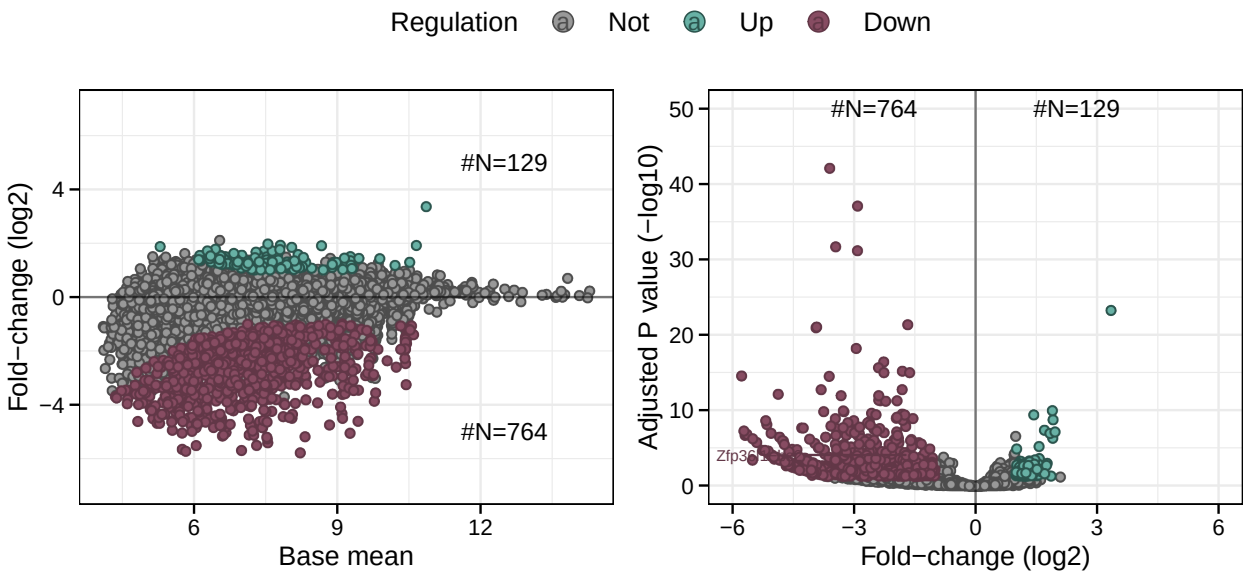


Figure 9: MA and Volcano plots. With adjusted P value and fold-change cutoffs. Custom annotaitons - significant only.

```
# MA + Volcano plots
df = diff_bs %>%
    mutate(sig = factor(ifelse(resBs.padj < 0.05 & abs(resBs.log2FoldChange) > log2(2), TRUE, FALSE))) %
    mutate(dir = factor(ifelse(resBs.log2FoldChange > 0, "Up", "Down"), level = c("Up", "Down"))) %>%
    mutate(sigDir = ifelse(sig == TRUE & dir == "Up", "Up", ifelse(sig == TRUE & dir == "Down", "Down",
    mutate(sigDir = factor(sigDir, levels = c("Not", "Up", "Down"))) %>%
    arrange(sigDir)
p1 = ggplot(df, aes(x = log2(resBs.baseMean), y = resBs.log2FoldChange, color = sigDir, fill = sigDir))
    geom_point_rast(shape = 21, stroke = 0.5, size = 1.5) +
    scale_fill_nice_udn_b() +
    scale_color_nice_udn_d() +
    geom_hline(yintercept = 0, color = "black", alpha = .5) +
    theme_nice() +
    guides(color = guide_legend(override.aes = list(size = 4))) +
    theme(legend.key.size = unit(1, 'cm'), legend.position = "top") +
    labs(
        x = "Base mean",
        y = "Fold-change (log2)",
```

```
        color = "Regulation",
        fill = "Regulation") +
    facet_wrap(~region)

p2 = ggplot(df, aes(x = resBs.log2FoldChange, y = -log10(resBs.padj), color = sigDir, fill = sigDir)) +
    geom_point_rast(shape = 21, stroke = 0.5, size = 1.5) +
    scale_fill_nice_udn_b() +
    scale_color_nice_udn_d() +
    geom_vline(xintercept = 0, color = "black", alpha = .5) +
    theme_nice() +
    guides(color = guide_legend(override.aes = list(size = 4))) +
    theme(legend.key.size = unit(1, 'cm'), legend.position = "top") +
    labs(
        x = "Fold-change (log2)",
        y = "Adjusted P value (-log10)",
        color = "Regulation",
        fill = "Regulation") +
    facet_wrap(~region)

p1 + plot_layout(guides = "collect") & theme(legend.position = "top")
```

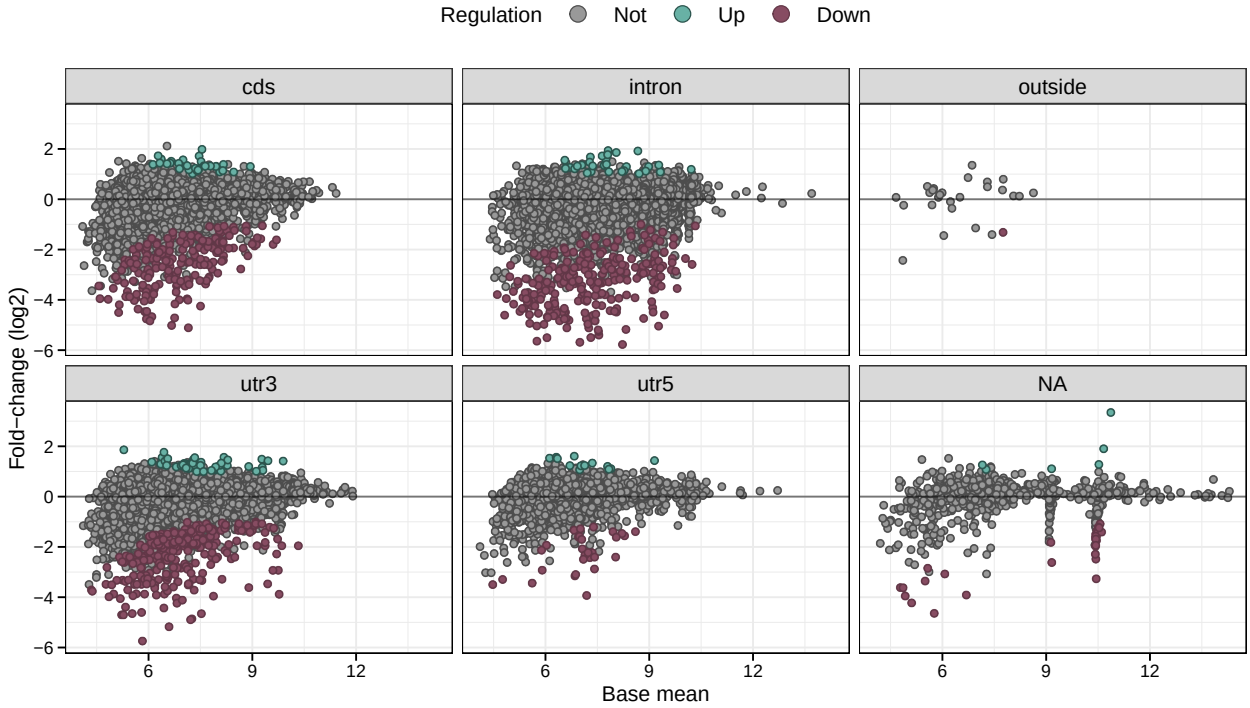


Figure 10: MA and Volcano plots. With adjusted P value and fold-change cutoffs. Split by region.

```
p2 + plot_layout(guides = "collect") & theme(legend.position = "top")
```

5.1.3 Export results

```
# export results
saveRDS(diff_bs, file = paste0(out, "BsDifferentialResult.rds"))
```

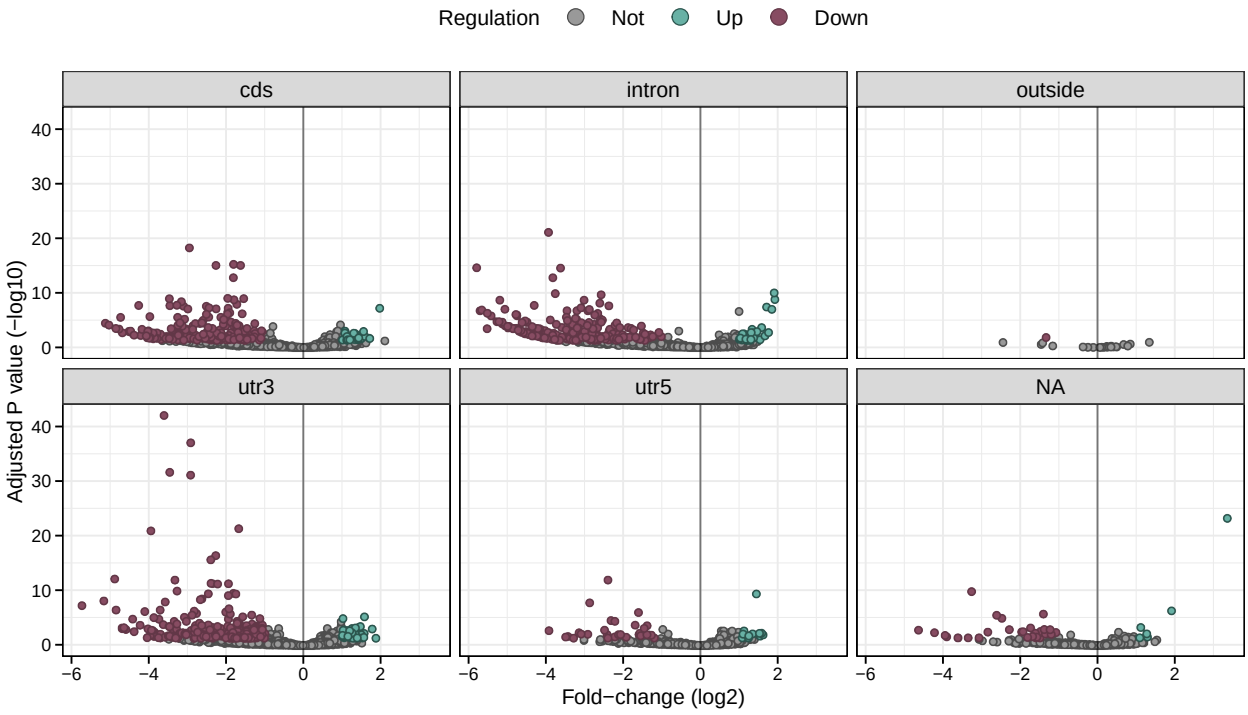


Figure 11: MA and Volcano plots. With adjusted P value and fold-change cutoffs. Split by region.

6 Session Info

```
sessionInfo()
```

```
## R version 4.2.2 (2022-10-31)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur ... 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] grid      stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] IHW_1.26.0                  DESeq2_1.38.3
##  [3] SummarizedExperiment_1.28.0 MatrixGenerics_1.10.0
##  [5] matrixStats_0.63.0          ggrastr_1.0.1
##  [7] ggsci_2.9                   ggpointdensity_0.1.0
##  [9] tidyr_1.3.0                 tibble_3.1.8
## [11] patchwork_1.1.2             ggtext_0.1.2
## [13] forcats_0.5.2               ComplexHeatmap_2.14.0
## [15] BindingSiteFinder_1.4.0     viridis_0.6.2
```

```
## [17] viridisLite_0.4.1          gridExtra_2.3
## [19] ggrepel_0.9.2              kableExtra_1.3.4
## [21] GenomicFeatures_1.50.4     UpSetR_1.4.0
## [23] reshape2_1.4.4             dplyr_1.0.10
## [25] AnnotationDbi_1.60.0       Biobase_2.58.0
## [27] ggplot2_3.4.0              rtracklayer_1.58.0
## [29] GenomicRanges_1.50.2       GenomeInfoDb_1.34.7
## [31] IRanges_2.32.0             S4Vectors_0.36.1
## [33] BiocGenerics_0.44.0        knitr_1.42
##
## loaded via a namespace (and not attached):
##   [1] utf8_1.2.2              tidyselect_1.2.0        RSQLite_2.2.20
##   [4] htmlwidgets_1.6.1       BiocParallel_1.32.5     munsell_0.5.0
##   [7] ragg_1.2.5              codetools_0.2-18        interp_1.1-3
##  [10] withr_2.5.0             colorspace_2.1-0        filelock_1.0.2
##  [13] highr_0.10              rstudioapi_0.14         ggsignif_0.6.4
##  [16] labeling_0.4.2          slam_0.1-50             GenomeInfoDbData_1.2.9
##  [19] lpsymphony_1.26.3       mixsqp_0.3-48           polyclip_1.10-4
##  [22] bit64_4.0.5             farver_2.1.1            vctrs_0.5.2
##  [25] generics_0.1.3          xfun_0.36               biovizBase_1.46.0
##  [28] BiocFileCache_2.6.0     R6_2.5.1                doParallel_1.0.17
##  [31] ggbeeswarm_0.7.1        clue_0.3-63             invgamma_1.1
##  [34] locfit_1.5-9.7          AnnotationFilter_1.22.0 bitops_1.0-7
##  [37] cachem_1.0.6            DelayedArray_0.24.0     assertthat_0.2.1
##  [40] BiocIO_1.8.0            scales_1.2.1            nnet_7.3-18
##  [43] beeswarm_0.4.0          gtable_0.3.1            Cairo_1.6-0
##  [46] ensembldb_2.22.0        rlang_1.0.6             systemfonts_1.0.4
##  [49] GlobalOptions_0.1.2     splines_4.2.2           rstatix_0.7.1
##  [52] lazyeval_0.2.2          dichromat_2.0-0.1       broom_1.0.3
##  [55] checkmate_2.1.0         abind_1.4-5             yaml_2.3.7
##  [58] backports_1.4.1         Hmisc_4.7-2             gridtext_0.1.5
##  [61] tools_4.2.2             ellipsis_0.3.2          RColorBrewer_1.1-3
##  [64] Rcpp_1.0.10             plyr_1.8.8              base64enc_0.1-3
##  [67] progress_1.2.2          zlibbioc_1.44.0         purrr_1.0.1
##  [70] RCurl_1.98-1.9          prettyunits_1.1.1       ggpubr_0.5.0
##  [73] rpart_4.1.19            deldir_1.0-6            GetoptLong_1.0.5
##  [76] ashr_2.2-54             cluster_2.1.4           magrittr_2.0.3
##  [79] magick_2.7.3            data.table_1.14.6       circlize_0.4.15
##  [82] truncnorm_1.0-9         SQUAREM_2021.1          ProtGenerics_1.30.0
##  [85] hms_1.1.2               evaluate_0.20           xtable_1.8-4
##  [88] XML_3.99-0.13           jpeg_0.1-10             shape_1.4.6
##  [91] compiler_4.2.2          biomaRt_2.54.0          crayon_1.5.2
##  [94] htmltools_0.5.4         Formula_1.2-4           geneplotter_1.76.0
##  [97] DBI_1.1.3               tweenr_2.0.2            dbplyr_2.3.0
## [100] MASS_7.3-58.2           rappdirs_0.3.3          car_3.1-1
## [103] Matrix_1.5-3            cli_3.6.0               parallel_4.2.2
## [106] Gviz_1.42.0             pkgconfig_2.0.3         GenomicAlignments_1.34.0
## [109] foreign_0.8-84          xml2_1.3.3              foreach_1.5.2
## [112] svglite_2.1.1           annotate_1.76.0         vipor_0.4.5
## [115] webshot_0.5.4           XVector_0.38.0          rvest_1.0.3
## [118] stringr_1.5.0           VariantAnnotation_1.44.0 digest_0.6.31
## [121] Biostrings_2.66.0       rmarkdown_2.20          htmlTable_2.4.1
## [124] restfulr_0.0.15         curl_5.0.0              Rsamtools_2.14.0
## [127] rjson_0.2.21            lifecycle_1.0.3         carData_3.0-5
```

```
## [130] BSgenome_1.66.2          fansi_1.0.4              pillar_1.8.1
## [133] lattice_0.20-45          KEGGREST_1.38.0          fastmap_1.1.0
## [136] httr_1.4.4               survival_3.5-0           glue_1.6.2
## [139] fdrtool_1.2.17           png_0.1-8                iterators_1.0.14
## [142] bit_4.0.5                ggforce_0.4.1            stringi_1.7.12
## [145] blob_1.2.3               textshaping_0.3.6        latticeExtra_0.6-30
## [148] memoise_2.0.1            irlba_2.3.5.1
```