mir181 binding sites - union of mir181 enriched binding sites and Ago binding sites targeted by mir181

Melina Klostermann

26 April, 2023

Contents

1 Libraries and settings

```
# ----------------------------------------
# libraries
# ----------------------------------------
library(tidyverse)
library(GenomicRanges)
library(colorspace)
library(eulerr)
library(gghalves)
library(ggpubr)

# ----------------------------------------
# settings
# ----------------------------------------
out <- "/Users/melinaklostermann/Documents/projects/AgoCLIP_miR181/R_github/miR181_paper/Figure1/mir181_
source("/Users/melinaklostermann/Documents/projects/R_general_functions/theme_paper.R")
source("/Users/melinaklostermann/Documents/projects/AgoCLIP_miR181/mirko_files/mirECLIP/DifferentialBin


# farben
farbeneg <- "#B4B4B4"
farbe1 <- "#0073C2FF" #WT farbe
farbe2 <- "#EFC000FF"
```

```
farbe3 <- "#CD534CFF" #miR181KO farbe
farbe4 <- "#7AA6DCFF"
farbe5 <- "#868686FF"
farbe6 <- "#003C67FF"
farbe7 <- "#8F7700FF"
farbe8 <- "#3B3B3BFF"
farbe9 <- "#A73030FF"
farbe10 <- "#4A6990FF"
farbe11 <- "#FF6F00FF"
farbe12 <- "#C71000FF"
farbe13 <- "#008EA0FF"
farbe14 <- "#8A4198FF"
farbe15 <- "#5A9599FF"
farbe16 <- "#FF6348FF"
```

2 What was done?

mir181 binding sites are defined as the union of - AGO binding sites that contain at least 2 chimirc mir181 crosslinks (from the IP__WT chimeric reads or the IP__mir181__WT chimeric reads) in a window from 10nt before till 10nt after a the AGO binding site - binding sites defined on enriched mir181 data (IP__mir181__WT)

- the two subgroups are plotted as a venn diagram (figure 1 XX)
- this is compared to the differntially regulated AGO binding sites from the mir181 KO condition

3 Files

```
# ----------------------------------------
# mir181 enriched binding sites
# ----------------------------------------
mir181_enriched <- readRDS("/Users/melinaklostermann/Documents/projects/AgoCLIP_miR181/R_github/miR181_p

# ----------------------------------------
# chimeric reads
# ----------------------------------------

chimeric_reads <- readRDS("/Users/melinaklostermann/Documents/projects/AgoCLIP_miR181/R_github/miR181_pa

# ----------------------------------------
# AGO binding sites
# ----------------------------------------
ago_bs <- readRDS("/Users/melinaklostermann/Documents/projects/AgoCLIP_miR181/R_github/miR181_paper/Fig

# ----------------------------------------
# BS downregulated in mir181 KO
# ----------------------------------------
diff <- readRDS("/Users/melinaklostermann/Documents/projects/AgoCLIP_miR181/R_github/miR181_paper/Figur
diff$down <- (diff$resBs.padj < 0.05) & (diff$resBs.log2FoldChange < -log2(2))
```

4 mir181 binding sites

4.1 Get AGO binding sites with chimeric mir181

Here we define mir181 AGO binding sites by overlapping the AGO binding sites (see script Methods/02_AGO_binding_site_definition) with the chimeric mir181 reads (see script Figure1/Ago_targetome). AGO binding sites that contained at least 2 chimeric mir181 crosslinks in the binding site or within 10nt proximity to the binding site are selected as mir181 Ago binding sites.

```
# use region of bs +-10nt for overlaps
ago_bs_10 <- ago_bs + 10

# use chimeric reads from both mir181 enriched and non-enriched data
chimeric_reads <- c(makeGRangesFromDataFrame(chimeric_reads$IP_WT, keep.extra.columns = T), makeGRangesF

# find overlaps of mirt and AGO bs
idx <- findOverlaps(ago_bs_10, chimeric_reads )

# make a data frame from the ago bs
names(ago_bs)<- 1:NROW(ago_bs)
ago_bs <- as.data.frame(ago_bs)
ago_bs$BS_ID <- rownames(ago_bs)

# add mir info to ago bs
ago_bs_mir181_chi <- cbind(ago_bs[queryHits(idx),], mir_IP = chimeric_reads [subjectHits(idx),]$Name)

ago_bs_mir181_chi <- ago_bs_mir181_chi[grepl(ago_bs_mir181_chi$mir_IP,
                                       pattern = "miR-181"),]

# count chimerics
mir181_chi <- ago_bs_mir181_chi %>% group_by(BS_ID) %>%
  summarize(n_mir181 = sum(grepl(mir_IP,pattern = "miR-181")),
            n_mir181a = sum(grepl(mir_IP,pattern = "miR-181a")),
            n_mir181b = sum(grepl(mir_IP,pattern = "miR-181b")),
            n_mir181c = sum(grepl(mir_IP,pattern = "miR-181c")),
            n_mir181d = sum(grepl(mir_IP,pattern = "miR-181d")),
            .groups = "keep") %>% subset (n_mir181 >0)

ago_bs_mir181_chi <- ago_bs_mir181_chi %>%
  subset(!duplicated(ago_bs_mir181_chi$BS_ID)) %>%
  left_join(., mir181_chi, by ="BS_ID") %>% makeGRangesFromDataFrame(keep.extra.columns = T)
```

4.2 Combine AGO binding sites with chimeric mir181 with mir181 enriched binding sites

I combine the mir181 Ago binding sites that we obtained above with the binding sites from the mir181 enriched Ago-eCLIP (see Methods/mir181-enriched_binding_site_definition). In order to do that, I first select binding sites from both conditions that do not overlap with any binding site from the other set. For the binding sites that overlap between the two conditions, I select the AGO mir181 binding sites and tag them as occuring in both sets. Then I combine the three subsets sets. The obtained union of mir181 binding sites from both conditions are our final mir181 binding sites.

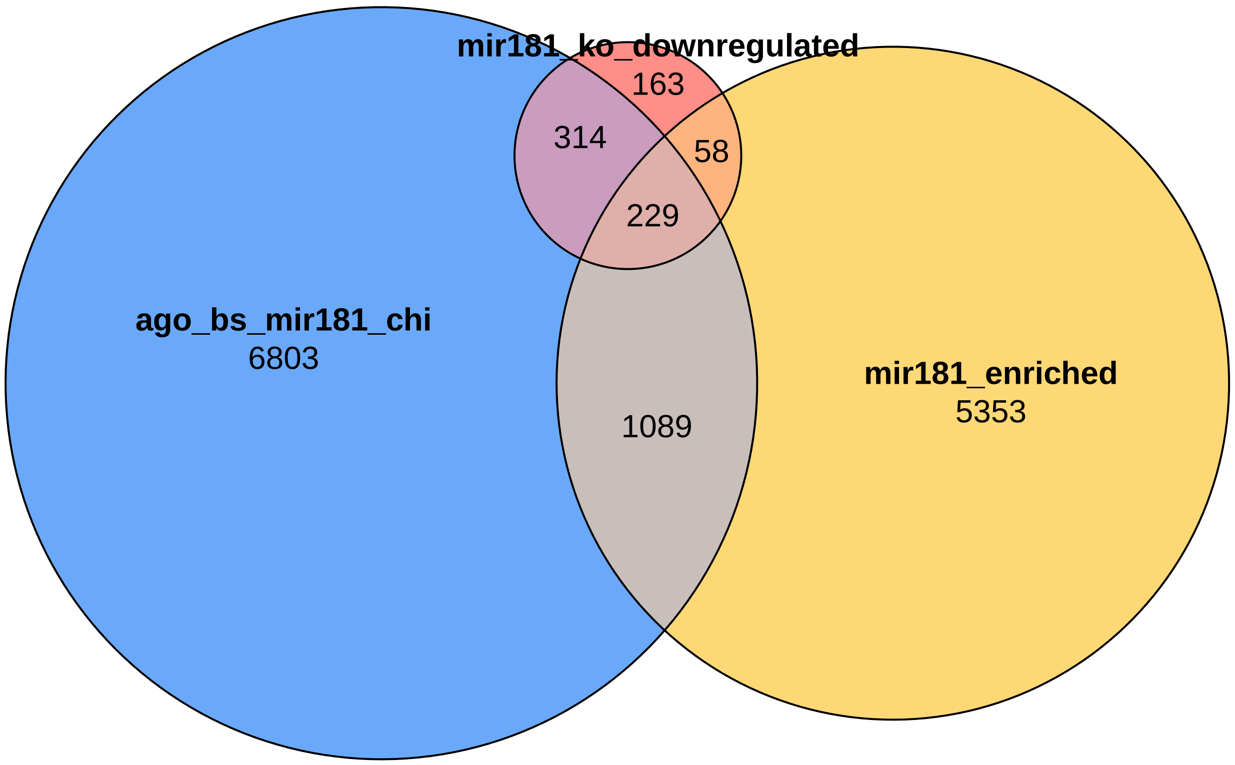
```
# ---------------------------
# combine mir181 Ago BS and mir181 enriched Bs
# --------------------------
```

```r
# get only Ago mir181 Bs with now overlaps to enriched mir181 BS
only_ago_bs_mir181_chi <- subsetByOverlaps(ago_bs_mir181_chi, mir181_enriched, type = "any", invert = T]
only_ago_bs_mir181_chi$set <- "ago_bs_mir181_chi"

# get only enriched mir181 BS with now overlaps to Ago mir181 Bs
only_mir181_enriched <- subsetByOverlaps(mir181_enriched, ago_bs_mir181_chi,  type = "any", invert = T)
only_mir181_enriched$set <- "mir181_enriched"

# get only Ago mir181 BS overlapping with mir181 enriched BS
both_mir181_enriched_chi <- subsetByOverlaps(ago_bs_mir181_chi, mir181_enriched, type = "any")
both_mir181_enriched_chi$set <- "ago_bs_mir181_chi&mir181_enriched"

# combine all three sets
mir181_bs <- c(only_ago_bs_mir181_chi, only_mir181_enriched, both_mir181_enriched_chi)
mir181_bs$BS_ID <- NULL
mir181_bs$mir181BS_ID <- 1:NROW(mir181_bs)
```

5 Combine mir181 binding sites with differntial binding sites

Next, I combine the obtained mir181 binding sites with the results we obtained from the differntial binding between AGO binding sites and AGO binding sites with mir181 KO (see script Figure1/Differntial_Binding_AGO_BS_mir181_KO).

```r
# ---------------------------
# combine with differential BS
# --------------------------
# get overlaps with diff binding
diff_overlap <- findOverlaps( mir181_bs, makeGRangesFromDataFrame(diff, keep.extra.columns = T) , type =
# add differential information to mir181 binding sites
d <- diff[,9:49]
mcols(mir181_bs) <- cbind(mcols(mir181_bs), d[diff_overlap,])

# add only diff bs (these are Ago binding sites but not mir181 Binding sites)
diff_only <- subsetByOverlaps( makeGRangesFromDataFrame(diff, keep.extra.columns = T), mir181_bs , type
mir181_bs_diff <- c(mir181_bs, diff_only)


# ---------------------------
# make venn diagram
# --------------------------

# select downregulated BS from differential BS
mir181_bs_diff_anygroup <- mir181_bs_diff %>% subset((down == T) | !is.na(set))

# make venn
venn_df <- data.frame(ago_bs_mir181_chi =
                        ((mir181_bs_diff_anygroup$set == "ago_bs_mir181_chi") | (mir181_bs_diff_anygrou
                      mir181_enriched =
                        ((mir181_bs_diff_anygroup$set == "mir181_enriched") | (mir181_bs_diff_anygroup$
                      mir181_ko_downregulated = (mir181_bs_diff_anygroup$down == T )
                      ) %>%
  mutate(ago_bs_mir181_chi = case_when(is.na(ago_bs_mir181_chi) ~ F, T~ago_bs_mir181_chi),
         mir181_enriched = case_when(is.na(mir181_enriched) ~ F, T ~ mir181_enriched),
        mir181_ko_downregulated = case_when(is.na(mir181_ko_downregulated) ~ F, T ~ mir181_ko_downregul



venn <- eulerr::euler(venn_df)
p <- plot(venn, quantities = T, fills = c( lighten(farbe1, amount = 0.4), lighten(farbe2, amount = 0.4)
p
```



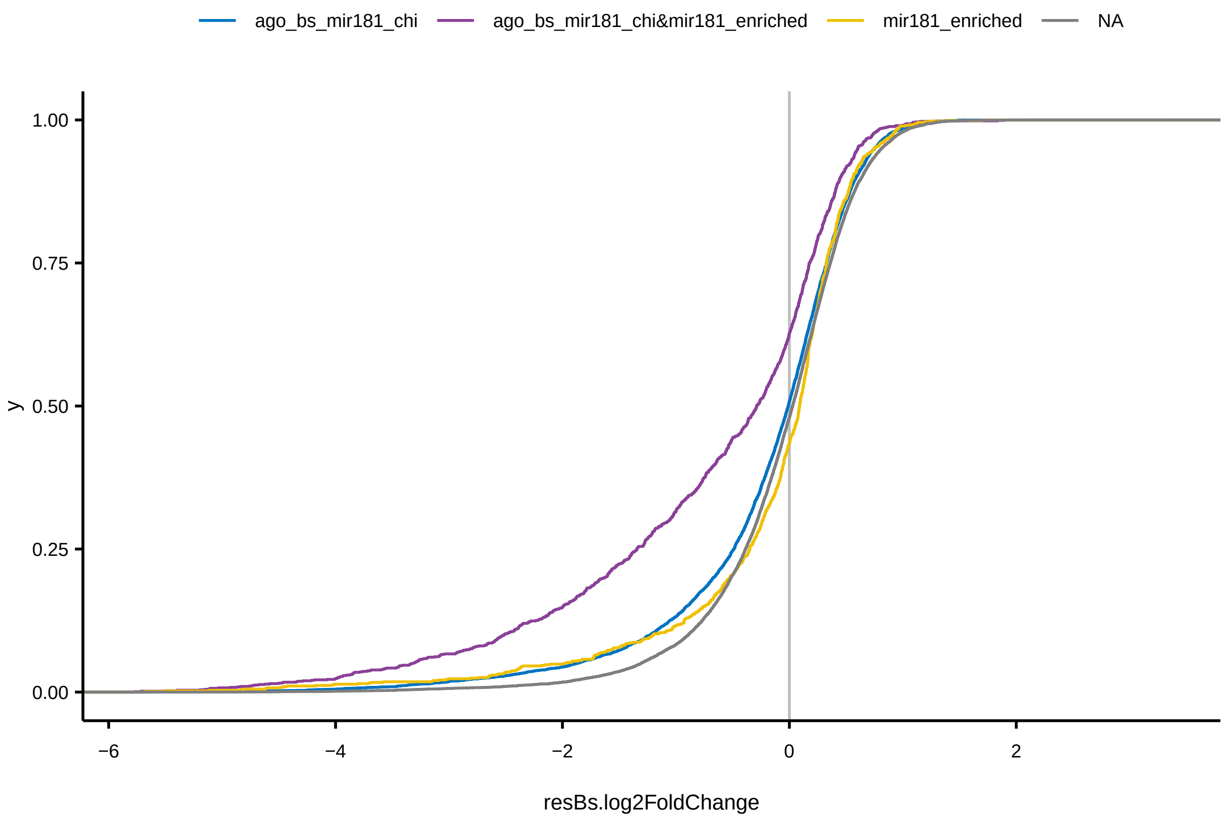
```
pdf(paste0(out, "Figure1I_Venn_overlaps_mirBs.pdf"), height = 3, width = 4 )
p
dev.off()
```

```
## pdf
##   2
```

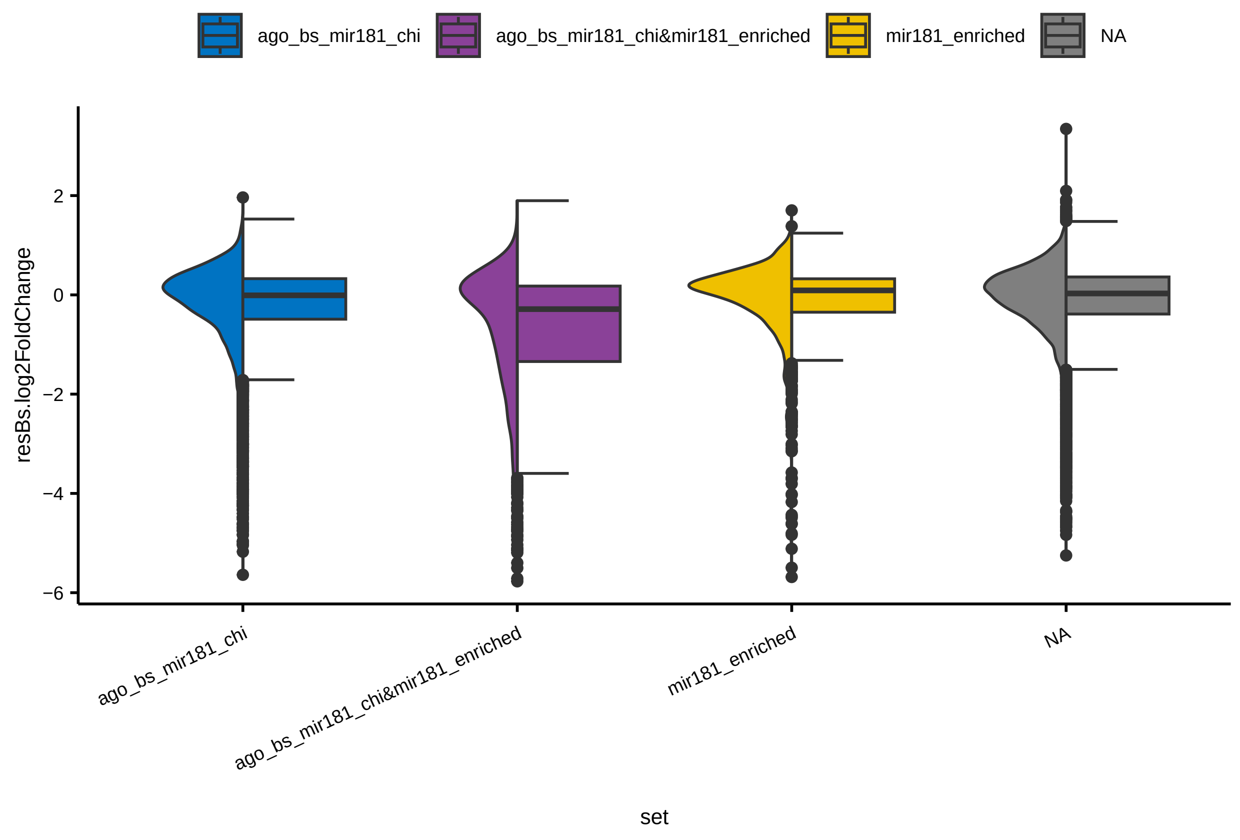
5.1 Differential binding both sets

Here I look at the overall binding changes between mir181 KO and WT for the three subsets of the mir181 binding sites.

```
# ---------------------------
# Compare regulation of BS during mir181 KO of the different subgroups of mir181 binding sites
# --------------------------
names(mir181_bs_diff) <- 1:NROW(mir181_bs_diff)

mir181_bs_diff <- as.data.frame(mir181_bs_diff)



p1 <- ggplot(mir181_bs_diff, aes(x = resBs.log2FoldChange, color = set))+
  geom_vline(xintercept = 0, color = "grey")+
  stat_ecdf()+
  scale_color_manual(values = c(farbe1, farbe14, farbe2, farbeneg))+
  theme_paper()+
  theme(legend.position = "top")


p2 <- ggplot(mir181_bs_diff, aes(y = resBs.log2FoldChange, x = set, fill = set))+
  geom_half_violin()+
  geom_half_boxplot(side = "r")+
  theme_paper()+
  scale_fill_manual(values = c(farbe1, farbe14, farbe2, farbeneg))+
  theme(legend.position = "top")+
  scale_x_discrete(guide = guide_axis(angle = 25))

p1
```



```
p2
```



```
ggsave(p1, filename =  paste0(out, "Figure_1J_ecdf_differntial_binding_vs_mir181BS.pdf"), width = 6, he
ggsave(p2, filename = paste0(out, "violin_differntial_binding_vs_mir181BS.pdf"), width = 10, height = 1
```

5.1.1 Statistical tests for differential binding changes

```
# t.tests of 3 sets against not bound
# ------------------------------------
t1 <- t.test(x = mir181_bs_diff %>% subset(set == "mir181_enriched") %>% pull(resBs.lfcSE),
       y = mir181_bs_diff %>% subset(is.na(set)) %>% pull(resBs.lfcSE))
t1
```

5.1.1.1 T-tests

```
##
##  Welch Two Sample t-test
##
## data:  mir181_bs_diff %>% subset(set == "mir181_enriched") %>% pull(resBs.lfcSE) and mir181_bs_diff %
## t = -15.992, df = 1000.5, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1305064 -0.1019782
## sample estimates:
## mean of x mean of y
## 0.4005421 0.5167844
```

```
t.test(x = mir181_bs_diff %>% subset(set == "ago_bs_mir181_chi") %>% pull(resBs.lfcSE),
       y = mir181_bs_diff %>% subset(is.na(set)) %>% pull(resBs.lfcSE))
```

```
##
```

```
##  Welch Two Sample t-test
##
## data:  mir181_bs_diff %>% subset(set == "ago_bs_mir181_chi") %>% pull(resBs.lfcSE) and mir181_bs_dif
## t = -17.36, df = 10729, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.05261416 -0.04193797
## sample estimates:
## mean of x mean of y
## 0.4695083 0.5167844
```

```
t.test(x = mir181_bs_diff %>% subset(set == "ago_bs_mir181_chi&mir181_enriched") %>% pull(resBs.lfcSE),
       y = mir181_bs_diff %>% subset(is.na(set)) %>% pull(resBs.lfcSE))
```

```
##
##  Welch Two Sample t-test
##
## data:  mir181_bs_diff %>% subset(set == "ago_bs_mir181_chi&mir181_enriched") %>% pull(resBs.lfcSE) ar
## t = -5.9647, df = 1276.1, p-value = 3.168e-09
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.06323341 -0.03193289
## sample estimates:
## mean of x mean of y
## 0.4692012 0.5167844
```

```
# --> the p-values are driven strongly by the number of binding sites per set, for this reason the high

# check power of test
# ------------------------------------
pwr::pwr.t2n.test(n1 = mir181_bs_diff %>% subset(set == "mir181_enriched") %>% nrow(.),
                  n2 = mir181_bs_diff %>% subset(is.na(set)) %>% nrow(.),
                  d = abs(t1$estimate[1] -t1$estimate[2])/t1$stderr)
```

```
##
##      t test power calculation
##
##              n1 = 5411
##              n2 = 17499
##               d = 15.99168
##       sig.level = 0.05
##           power = 1
##     alternative = two.sided
```

```

# Kolmogorov-Smirnov Tests
# -------------------------
ks.test(x = mir181_bs_diff %>% subset(set == "mir181_enriched") %>% arrange(resBs.lfcSE) %>%pull(resBs.l
        y = mir181_bs_diff %>% subset(is.na(set) ) %>% arrange(resBs.lfcSE) %>%pull(resBs.lfcSE))
```

5.1.1.2 Kolmogorov-Smirnov Tests

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  mir181_bs_diff %>% subset(set == "mir181_enriched") %>% arrange(resBs.lfcSE) %>% pull(resBs.l
## D = 0.4051, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
ks.test(x = mir181_bs_diff %>% subset(set == "ago_bs_mir181_chi") %>% arrange(resBs.lfcSE) %>%pull(resBs
        y = mir181_bs_diff %>% subset(is.na(set) ) %>% arrange(resBs.lfcSE) %>%pull(resBs.lfcSE))
```

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  mir181_bs_diff %>% subset(set == "ago_bs_mir181_chi") %>% arrange(resBs.lfcSE) %>% pull(resBs
## D = 0.17649, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

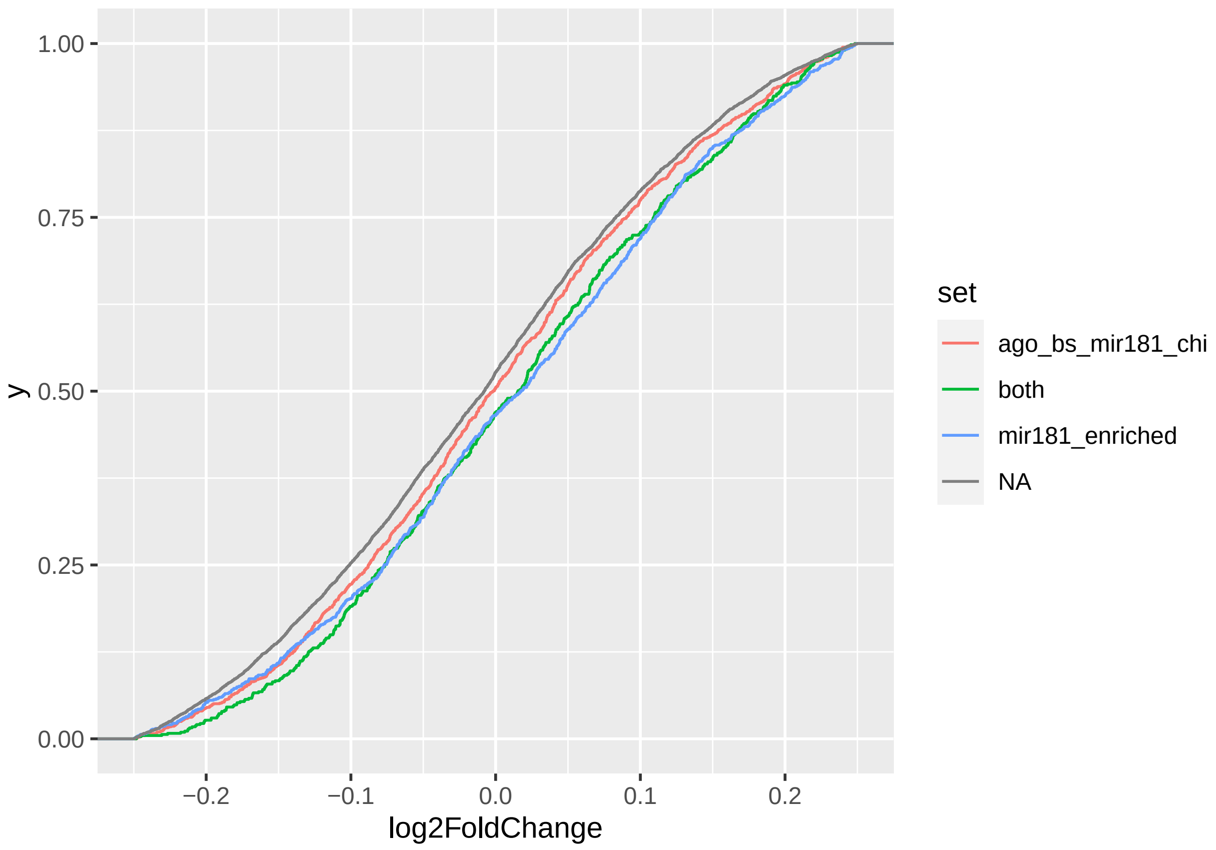
```
ks.test(x = mir181_bs_diff %>% subset(set == "ago_bs_mir181_chi&mir181_enriched") %>% arrange(resBs.lfcS
        y = mir181_bs_diff %>% subset(is.na(set) ) %>% arrange(resBs.lfcSE) %>%pull(resBs.lfcSE))
```

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  mir181_bs_diff %>% subset(set == "ago_bs_mir181_chi&mir181_enriched") %>% arrange(resBs.lfcSE
## D = 0.3015, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

5.2 Venn bound genes from all sets

5.3 Ribofootprint both sets

```
# TODO need to be other script later
rpf <- read.csv("/Users/melinaklostermann/Documents/projects/AgoCLIP_miR181/files_RNAseq_RiboP/RPF_maste
rpf <- mutate(rpf, Gene = substr(Gene,1,18))

rpf <- rpf %>% mutate(., set =
                        case_when(Gene %in% both_mir181_enriched_chi$geneID ~ "both",
                                  (Gene %in% ago_bs_mir181_chi$geneID) & !(Gene %in% both_mir181_enriche
                                  (Gene %in% mir181_enriched$geneID) & !(Gene %in% both_mir181_enriched
                                  ))

ggplot(rpf, aes(x = log2FoldChange, color = set))+
  stat_ecdf()+
  xlim(-0.25, 0.25)
```



6 Save output

```
saveRDS(mir181_bs, paste0(out, "mir181_bs.rds"))
```

7 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] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] ggpubr_0.5.0         gghalves_0.1.4       eulerr_7.0.0
##  [4] colorspace_2.1-0     GenomicRanges_1.50.2 GenomeInfoDb_1.34.7
##  [7] IRanges_2.32.0       S4Vectors_0.36.1     BiocGenerics_0.44.0
## [10] forcats_0.5.2        stringr_1.5.0        dplyr_1.0.10
## [13] purrr_1.0.1          readr_2.1.3          tidyr_1.3.0
## [16] tibble_3.1.8         ggplot2_3.4.0        tidyverse_1.3.2
## [19] knitr_1.42
##
## loaded via a namespace (and not attached):
##  [1] httr_1.4.4             jsonlite_1.8.4         carData_3.0-5
##  [4] modelr_0.1.10          assertthat_0.2.1       highr_0.10
##  [7] GenomeInfoDbData_1.2.9 googlesheets4_1.0.1    cellranger_1.1.0
## [10] yaml_2.3.7             pillar_1.8.1           backports_1.4.1
## [13] glue_1.6.2             digest_0.6.31          XVector_0.38.0
## [16] ggsignif_0.6.4         polyclip_1.10-4        rvest_1.0.3
## [19] htmltools_0.5.4        pkgconfig_2.0.3        broom_1.0.3
## [22] haven_2.5.1            zlibbioc_1.44.0        scales_1.2.1
## [25] tzdb_0.3.0             timechange_0.2.0       googledrive_2.0.0
## [28] farver_2.1.1           generics_0.1.3         car_3.1-1
## [31] ellipsis_0.3.2         withr_2.5.0            cli_3.6.0
## [34] magrittr_2.0.3         crayon_1.5.2           readxl_1.4.1
## [37] evaluate_0.20          fs_1.6.0               fansi_1.0.4
## [40] rstatix_0.7.1          xml2_1.3.3             textshaping_0.3.6
## [43] tools_4.2.2            hms_1.1.2              gargle_1.2.1
## [46] lifecycle_1.0.3        munsell_0.5.0          reprex_2.0.2
## [49] compiler_4.2.2         systemfonts_1.0.4      rlang_1.0.6
## [52] grid_4.2.2             RCurl_1.98-1.9         rstudioapi_0.14
## [55] labeling_0.4.2         bitops_1.0-7           rmarkdown_2.20
## [58] gtable_0.3.1           abind_1.4-5            DBI_1.1.3
## [61] R6_2.5.1               lubridate_1.9.1        pwr_1.3-0
## [64] fastmap_1.1.0          utf8_1.2.2             ragg_1.2.5
## [67] polylabelr_0.2.0       stringi_1.7.12         Rcpp_1.0.10
## [70] vctrs_0.5.2            dbplyr_2.3.0           tidyselect_1.2.0
## [73] xfun_0.36
```