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

Melina Klostermann

06 April, 2023

Contents

1 Libraries and settings

```
# ----------------------------------------
# libraries
# ----------------------------------------
library(tidyverse)
library(GenomicRanges)

# ----------------------------------------
# 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 (TODO)
- the genetype and gene region of the mir 181 binding sites (union) are ploted (Figure2XX)

3 Files

```
# ----------------------------------------
# mir181 enriched binding sites
# ----------------------------------------
mir181_enriched <- readRDS("/Users/melinaklostermann/Documents/projects/AgoCLIP_miR181/R_github/miR181_

# ----------------------------------------
# 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/Met
```

4 mir181 binding sites

4.1 Get AGO binding sites with chimeric mir181

```
# 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
```

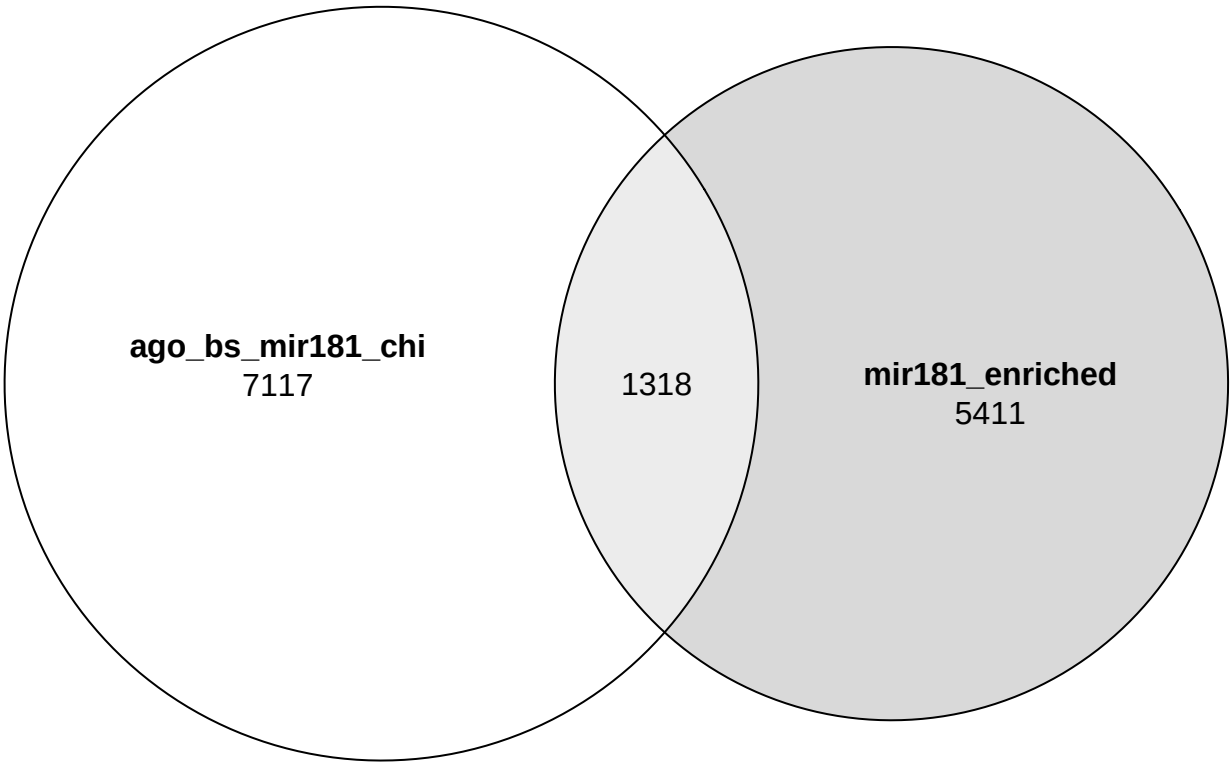
```
idx <- findOverlaps(ago_bs_10, chimeric_reads )

# 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

```
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"

only_mir181_enriched <- subsetByOverlaps(mir181_enriched, ago_bs_mir181_chi,  type = "any", invert = T)
only_mir181_enriched$set <- "mir181_enriched"

both_mir181_enriched_chi <- subsetByOverlaps(ago_bs_mir181_chi, mir181_enriched, type = "any")
both_mir181_enriched_chi$set <- "ago_bs_mir181_chi&mir181_enriched"


mir181_bs <- c(only_ago_bs_mir181_chi, only_mir181_enriched, both_mir181_enriched_chi)
```

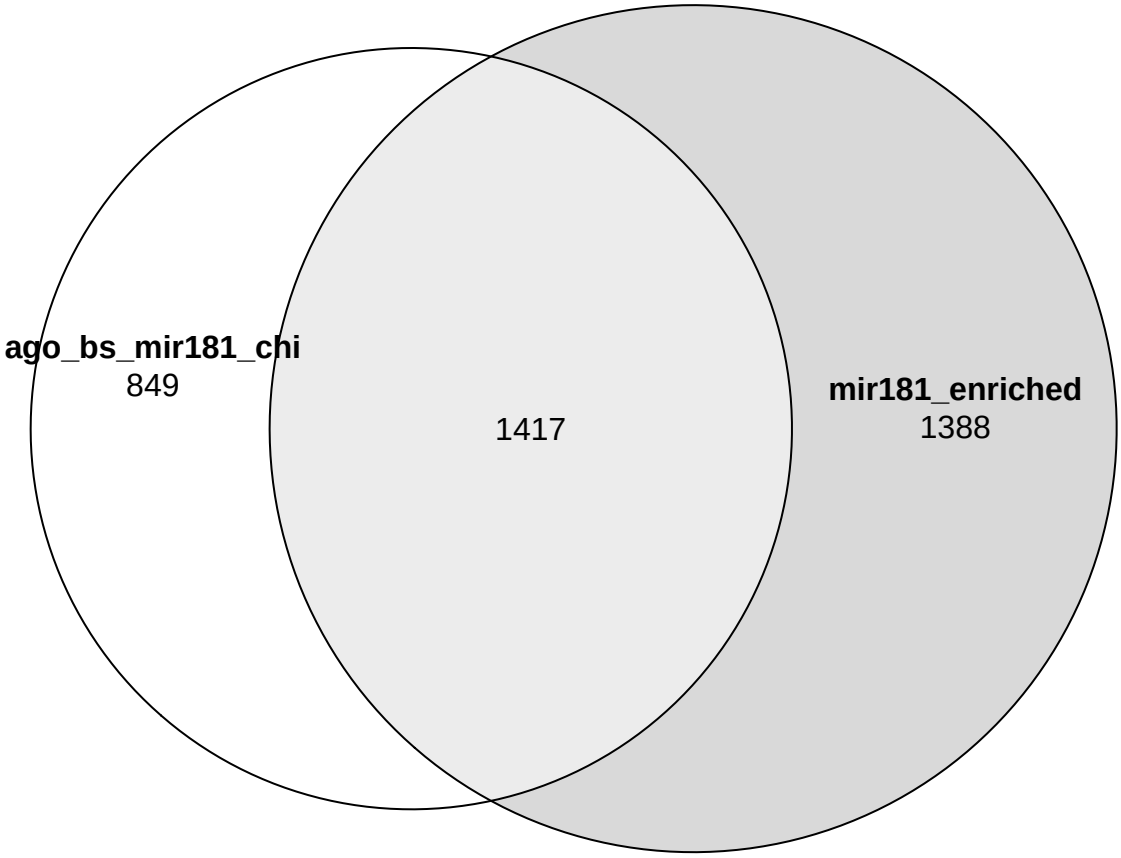
4.2.1 Venn binding sites from both sets

```
venn_df <- data.frame(ago_bs_mir181_chi =
                        ((mir181_bs$set == "ago_bs_mir181_chi") | (mir181_bs$set == "ago_bs_mir181_chi&m
                      mir181_enriched =
                        ((mir181_bs$set == "mir181_enriched") | (mir181_bs$set == "ago_bs_mir181_chi&mi

venn <- eulerr::euler(venn_df)
plot(venn, quantities = T)
```



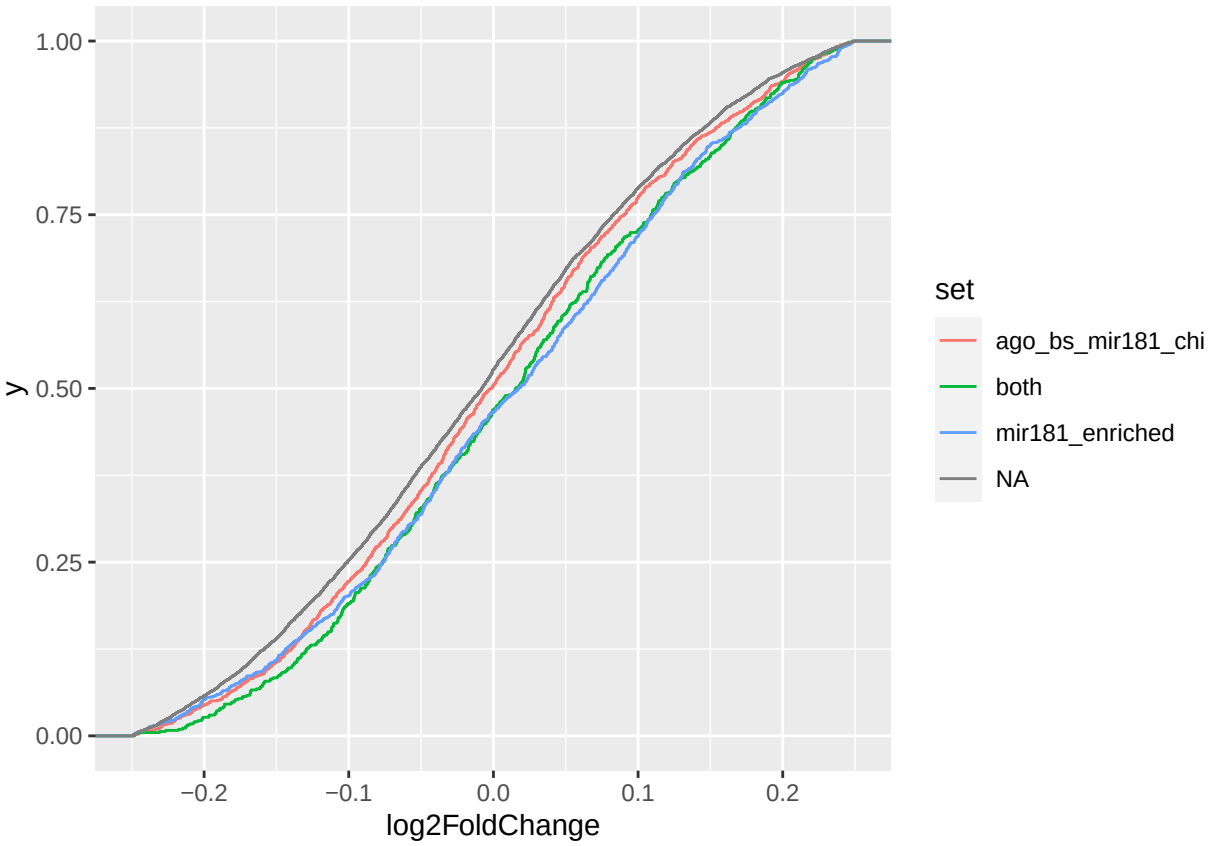
4.2.2 Venn bound genes from both sets

```
bound_genes <- unique(mir181_bs$geneID)
venn_df <- data.frame(ago_bs_mir181_chi =
                        (bound_genes %in% c(mir181_bs[mir181_bs$set == "ago_bs_mir181_chi"]$geneID,
                                            mir181_bs[mir181_bs$set == "ago_bs_mir181_chi&mir181_enriche
                      mir181_enriched =
                        (bound_genes %in% c(mir181_bs[mir181_bs$set == "mir181_enriched"]$geneID,
                                            mir181_bs[mir181_bs$set == "ago_bs_mir181_chi&mir181_enriche

venn <- eulerr::euler(venn_df)
plot(venn, quantities = T)
```



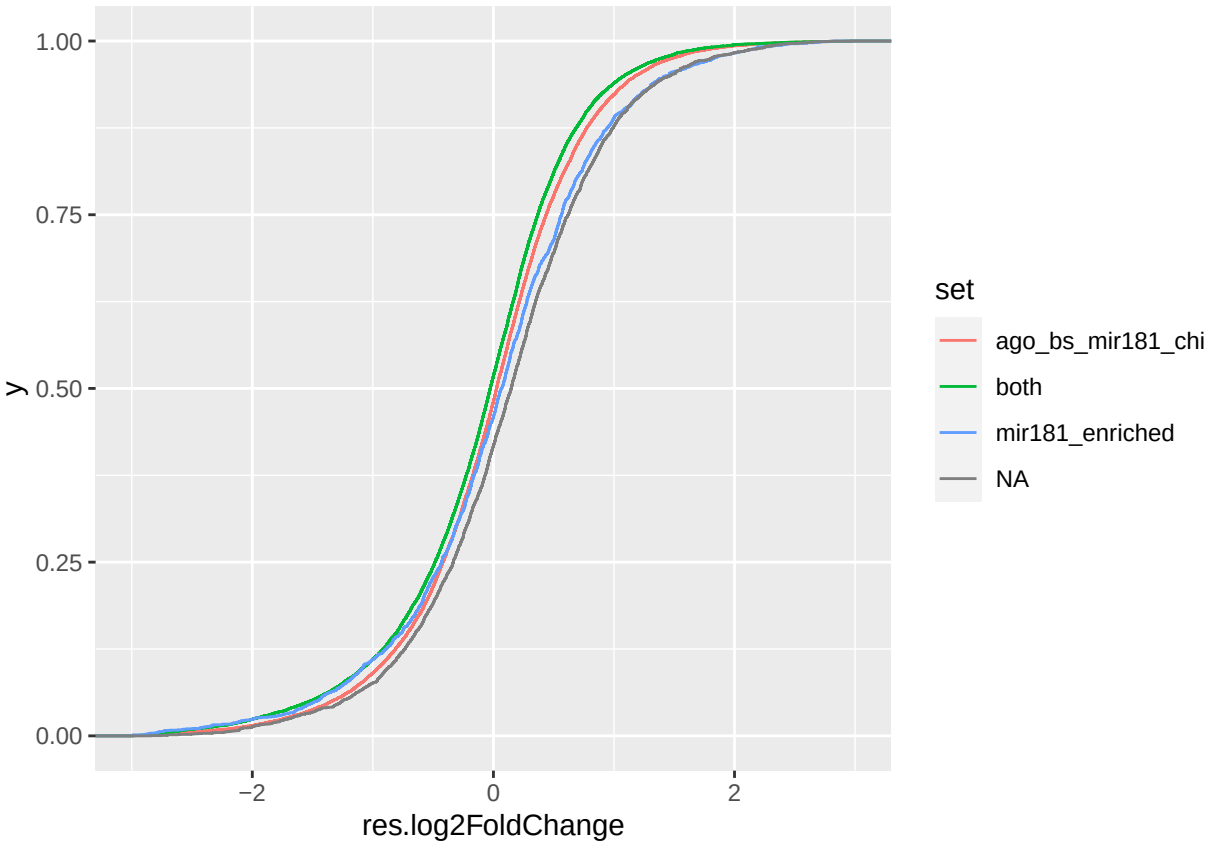
4.2.3 Ribofootprint both sets

```
# 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)
```



4.3 Differential binding both sets

```
diff <- readRDS("/Users/melinaklostermann/Documents/projects/AgoCLIP_miR181/mirko_files/mirECLIP/Differ

diff <- as.data.frame(diff) %>% mutate(., set =
                          case_when(geneID %in% both_mir181_enriched_chi$geneID ~ "both",
                                    (geneID %in% ago_bs_mir181_chi$geneID) & !(geneID %in% both_mir181_en
                                    (geneID %in% mir181_enriched$geneID) & !(geneID %in% both_mir181_enri
                                    ))

ggplot(diff, aes(x = res.log2FoldChange, color = set))+
  stat_ecdf()+
  xlim(-3,3)
```

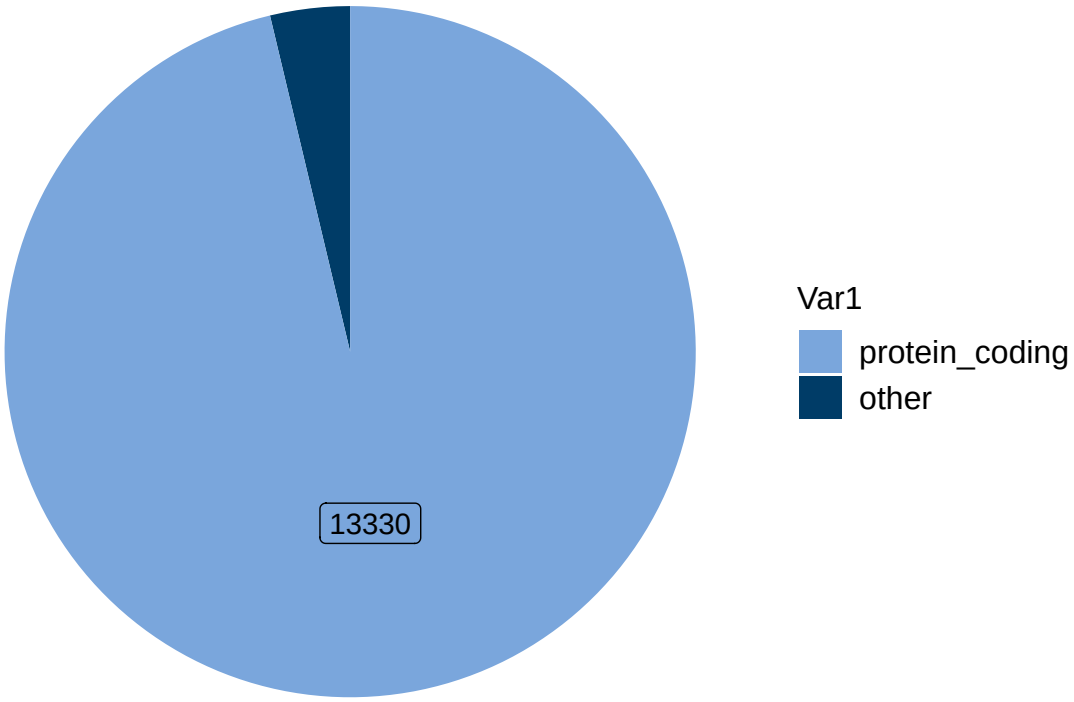


#

Characterise mir181 binding sites

4.4 mir181 bound genes - Figure 2A?

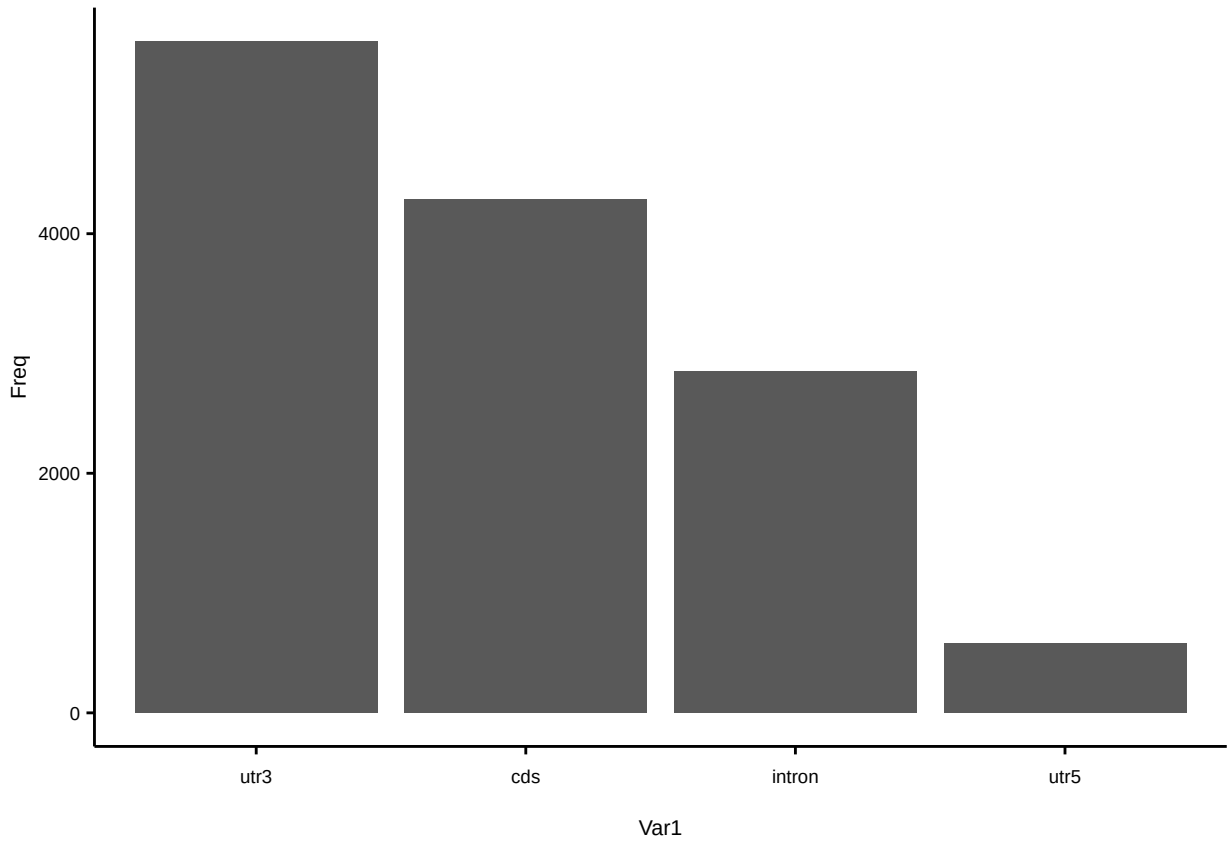
```
names(mir181_bs) <- 1:NROW(mir181_bs)
mir181_bs <- as.data.frame(mir181_bs)

gene_type_df <- mutate(mir181_bs, geneType = case_when(geneType != "protein_coding" ~ "other", T ~ "pro
gene_type_df <-   table(gene_type_df$geneType) %>%
  as.data.frame(.)

p <- ggplot(gene_type_df, aes(y=Freq, x="", fill=Var1)) +
     geom_col()+
     coord_polar(theta="y") +
  #    xlim(c(2, 4)) +
  geom_label(data = gene_type_df %>% subset(gene_type_df == "protein_coding"), aes(y=Freq, x="", fill=Va
             position = position_stack(vjust = 0.5),
             show.legend = FALSE) +
  scale_fill_manual(values = c (farbe6, farbe4)) +
  theme_paper() +
  theme_nice_pie() +
  #theme(legend.position = "none") +
  guides(fill = guide_legend(reverse = TRUE)) +
  labs(y = NULL,
       x = NULL)
p
```




```
ggsave(p, filename = paste0(out, "Figure2A_bound_gene_types_miR181_BS", Sys.Date(), ".pdf"), width = uni
```

4.5 mir181 bound regions - Figure 2B?

```
gene_region_df <- table(mir181_bs$region) %>%
  as.data.frame(.) %>%
  arrange(desc(Freq))

gene_region_df$Var1 <- factor(gene_region_df$Var1, levels = gene_region_df$Var1)

p <- ggplot(gene_region_df %>% subset(Var1 != "outside"), aes(y=Freq, x=Var1)) +
    geom_col()+
  theme_paper()
p
```




```
ggsave(p, filename = paste0(out, "Figure2B_bound_gene_regions_miR181_BS", Sys.Date(), ".pdf"), width = w
```

5 Save output

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

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] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
```

```
## other attached packages:
##  [1] GenomicRanges_1.50.2 GenomeInfoDb_1.34.7  IRanges_2.32.0
##  [4] S4Vectors_0.36.1     BiocGenerics_0.44.0  forcats_0.5.2
##  [7] stringr_1.5.0        dplyr_1.0.10         purrr_1.0.1
## [10] readr_2.1.3          tidyr_1.3.0          tibble_3.1.8
## [13] ggplot2_3.4.0        tidyverse_1.3.2      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] colorspace_2.1-0       eulerr_7.0.0           htmltools_0.5.4
## [22] pkgconfig_2.0.3        broom_1.0.3            haven_2.5.1
## [25] zlibbioc_1.44.0        scales_1.2.1           tzdb_0.3.0
## [28] timechange_0.2.0       googledrive_2.0.0      car_3.1-1
## [31] generics_0.1.3         farver_2.1.1           ellipsis_0.3.2
## [34] ggpubr_0.5.0           withr_2.5.0            cli_3.6.0
## [37] magrittr_2.0.3         crayon_1.5.2           readxl_1.4.1
## [40] evaluate_0.20          fs_1.6.0               fansi_1.0.4
## [43] rstatix_0.7.1          xml2_1.3.3             textshaping_0.3.6
## [46] tools_4.2.2            hms_1.1.2              gargle_1.2.1
## [49] lifecycle_1.0.3        munsell_0.5.0          reprex_2.0.2
## [52] compiler_4.2.2         systemfonts_1.0.4      rlang_1.0.6
## [55] grid_4.2.2             RCurl_1.98-1.9         rstudioapi_0.14
## [58] bitops_1.0-7           labeling_0.4.2         rmarkdown_2.20
## [61] gtable_0.3.1           abind_1.4-5            DBI_1.1.3
## [64] R6_2.5.1               lubridate_1.9.1        fastmap_1.1.0
## [67] utf8_1.2.2             ragg_1.2.5             polylabelr_0.2.0
## [70] stringi_1.7.12         Rcpp_1.0.10            vctrs_0.5.2
## [73] dbplyr_2.3.0           tidyselect_1.2.0       xfun_0.36
```