AGO targetome

Melina Klostermann

12 September, 2023

Contents

1 Libraries and settings

2 What was done?

- Overview of the detected chimeric reads in all conditions.
- Chimeric reads are assigned to AGO-binding sites (chimeric AGO sites).
- Co-ocurrence of miRs in the same AGO binding site (fisher-test heatmap).

We obtain chimeric reads from 4 different conditions: - AGO eCLIP (IP_WT) - AGO eCLIP with mir181a KO and mir181b KO (IP_KO) - AGO eCLIP with mir181 enrichment (IP_mir181_WT) - AGO eCLIP with mir181 enrichment and with mir181a KO and mir181b KO (IP_mir181_KO)

3 Files

```
# ----------------------------------------
# AGO binding sites
# ----------------------------------------
ago_bs <- readRDS(paste0(here,"/Figure1+SF1a-g/02_AGO_binding_site_definition/AGO_BS.rds"))


# ----------------------------------------
# chimeric reads
# ----------------------------------------
mir_crosslinks <- list.files("/Users/melinaklostermann/Documents/projects/AgoCLIP_miR181/pipe_output_22
```

```
                              pattern = "*.bed", recursive = TRUE, full.names = T) %>% map(~import.bed(.
names(mir_crosslinks) <- list.files("/Users/melinaklostermann/Documents/projects/AgoCLIP_miR181/pipe_ou
                              pattern = "*.bed")

# Nikita source
# mir_crosslinks <- list.files("D:/Krueger_Lab/miReCLIP/Melina/pipe_output_22_02_14/pipe_output_22_02_1
#                               pattern = "*.bed", recursive = TRUE, full.names = T) %>% map(~import.bed
# names(mir_crosslinks) <- list.files("D:/Krueger_Lab/miReCLIP/Melina/pipe_output_22_02_14/pipe_output_
#                               pattern = "*.bed")
#
#
#

# ----------------------------------------
# mir181 expression imgen
# ----------------------------------------
mir_imgen <- read.csv("/Users/melinaklostermann/Documents/projects/AgoCLIP_miR181/files_public_mir_data/

# Nikita source
# mir_imgen <- read.csv("D:/Krueger_Lab/Publications/miR181_paper/Figure1/Ago_targetome/DPsets_immgen.c
```

4 Chimeric reads

These are the chimeric reads that were isolated during the read processing via raccon (link TODO)

4.1 Number of chimeric reads per sample

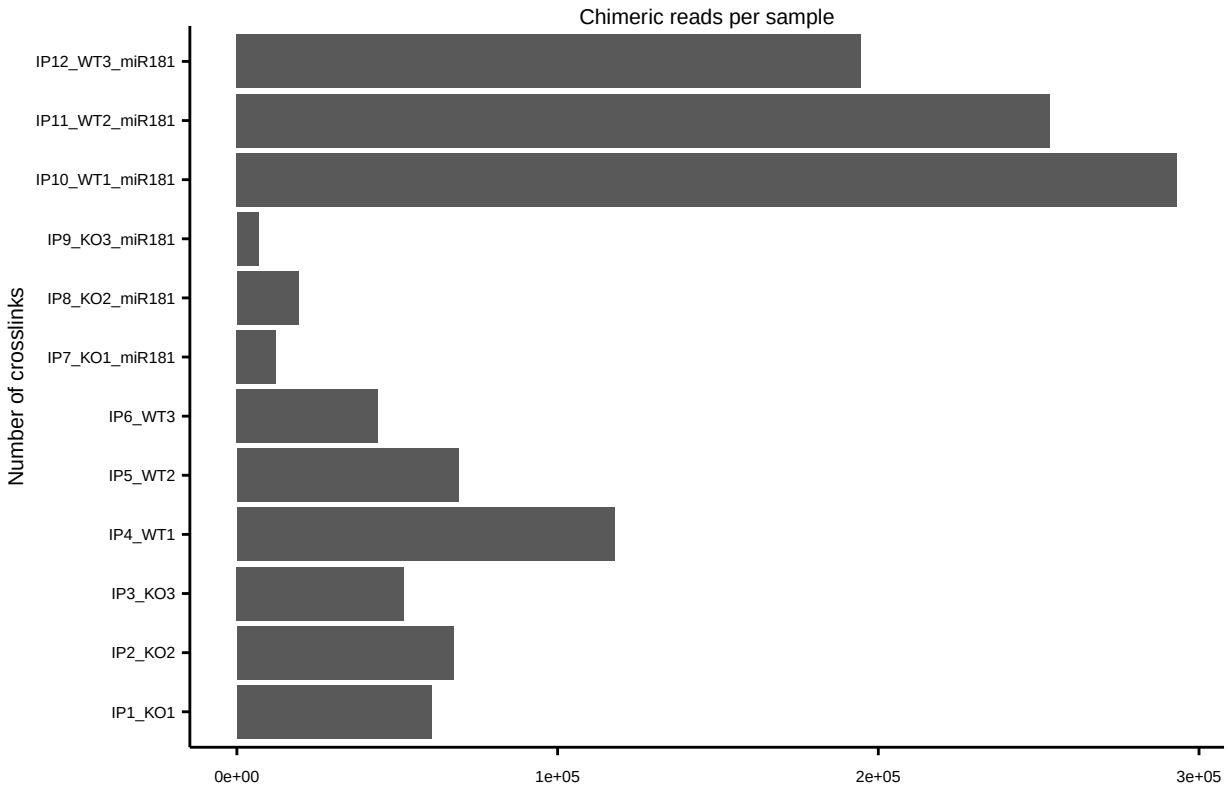
```
# ----------------------------------------
# get chimeric reads
# ----------------------------------------

# clean files
mir_crosslinks <- map(mir_crosslinks, ~ as.data.frame(.x) %>%
                        mutate(strand = Strand, Strand = NULL))

sample_names <- c("Inp1_KO1", "Inp2_KO2", "Inp3_KO3",
                  "Inp4_WT1", "Inp5_WT2", "Inp6_WT3",
                  "IP1_KO1", "IP2_KO2", "IP3_KO3",
                  "IP4_WT1", "IP5_WT2", "IP6_WT3",
                  "IP7_KO1_miR181", "IP8_KO2_miR181", "IP9_KO3_miR181",
                  "IP10_WT1_miR181", "IP11_WT2_miR181", "IP12_WT3_miR181"
                  )

mir_crosslinks <- map( sample_names , ~bind_rows(mir_crosslinks[grepl(names(mir_crosslinks), pattern =

names(mir_crosslinks) <- sample_names

# make overview table

table_num_crosslinks <- map_dfr(mir_crosslinks, ~c(number_of_crosslinks = NROW(.x)))
table_num_crosslinks$sample <- sample_names
```

sample	number_of_crosslinks
Inp1_KO1	946
Inp2_KO2	737
Inp3_KO3	717
Inp4_WT1	854
Inp5_WT2	951
Inp6_WT3	698
IP1_KO1	60,789
IP2_KO2	67,639
IP3_KO3	52,100
IP4_WT1	117,849
IP5_WT2	69,074
IP6_WT3	43,983
IP7_KO1_miR181	12,186
IP8_KO2_miR181	19,264
IP9_KO3_miR181	6,832
IP10_WT1_miR181	293,149
IP11_WT2_miR181	253,502
IP12_WT3_miR181	194,628

```
kable(table_num_crosslinks[,c(2,1)], format.args = list(big.mark = ",")) %>%
  kable_material(c("striped", "hover")) %>%
  scroll_box(width = "100%", height = "500px")
```

4.2 FigureS1B Barchart chimeric reads per sample

```
# ----------------------------------------
# Chimeric reads per sample
# Supplementary Figure 1 b
# ----------------------------------------


gg_df <- table_num_crosslinks %>% subset(sample %in% c("IP1_KO1", "IP2_KO2", "IP3_KO3",
                  "IP4_WT1", "IP5_WT2", "IP6_WT3",
                  "IP7_KO1_miR181", "IP8_KO2_miR181", "IP9_KO3_miR181",
                  "IP10_WT1_miR181", "IP11_WT2_miR181", "IP12_WT3_miR181"))

p <- ggplot(gg_df, aes(x = factor(sample, levels = sample), y = number_of_crosslinks))+
  geom_col()+
  coord_flip()+
  ylab("")+
  xlab("Number of crosslinks")

p + ggtitle("Chimeric reads per sample")
```




```
p <- p + theme_paper()

ggsave(p, filename = paste0(out, "FigureS1B_Barchart_chimeric_reads.pdf"), width = unit(6, "cm"), heigh
```

4.3 Detected mirs per sample

```
detected_mirs <- map(mir_crosslinks, ~.x %>%
                       group_by(`Name`) %>%
                       summarise(n = sum(Score), .groups= "keep") )

detected_mirs <- map(detected_mirs, ~arrange(.x, desc(n)))
```

4.4 Detected mirs per condition

```
condition_regex <- list("Inp.+KO", "Inp.+WT",
                  "IP.+KO[1-3]+$",  "IP.+WT[1-3]+$",
                  "IP.+KO.+_miR181", "IP.+_WT.+_miR181" )
condition_names <- list("Inp_KO", "Inp_WT",
                  "IP_KO",  "IP_WT",
                  "IP_KO_miR181", "IP_WT_miR181" )


mir_crosslinks_per_cond <- map(condition_regex,
                   ~bind_rows(mir_crosslinks[grepl(names(mir_crosslinks), pattern =.x)] ))
```

```
names(mir_crosslinks_per_cond ) <- condition_names
```

```
detected_mirs_per_cond <- map(mir_crosslinks_per_cond, ~.x %>%
                        group_by(`Name`) %>%
                        summarise(n = sum(Score), .groups= "keep",
                                  mean = n/3) )

detected_mirs_per_cond <- map(detected_mirs_per_cond , ~arrange(.x, desc(n)))

detected_mirs_per_cond_top_10 <- map(detected_mirs_per_cond, ~.x[1:10,] %>%
                                arrange(., n))
```

4.4.1 AGO-IP

- Number of differnt miRs in AGO-IP: 317
- number of chimeric reads in AGO-IP: 2.30906×10^5
- percent mir181a in AGO-IP: 0.2335236
- percent mir181a in AGO-IP: 0.2778143

4.4.2 mir181 AGO IP

- Number of differnt miRs in mir181-AGO-IP: 171
- number of chimeric reads in mir181-AGO-IP: 7.41279×10^5
- percent mir181a in mir181-AGO-IP: 0.7987829
- percent mir181a in mir181-AGO-IP: 0.1806364

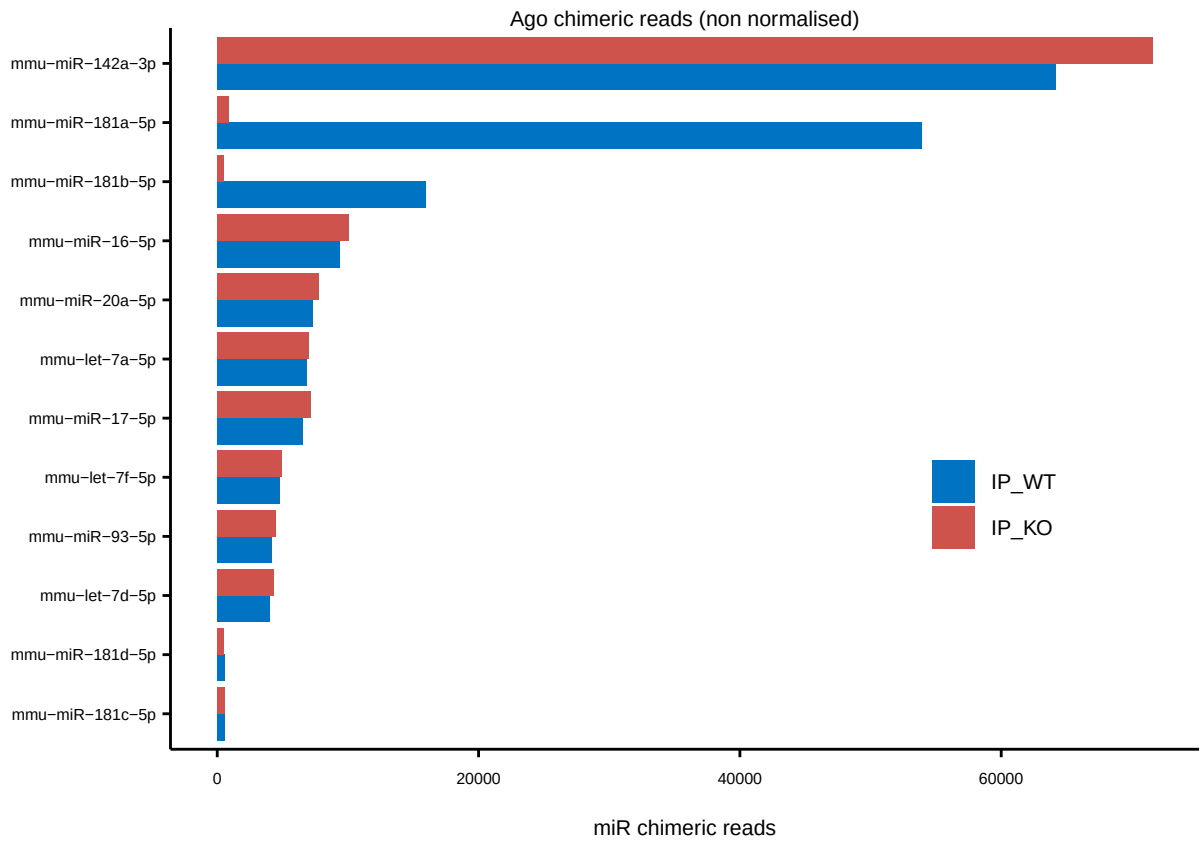
4.5 Supplementary Table 1

```
# ----------------------------------------
# excel tables of detected miRs
# Supplementary Table 1
# ----------------------------------------

xlsx::write.xlsx(x = as.data.frame(detected_mirs_per_cond$IP_WT), file = paste0(out,"detected_miR_IP_WT
xlsx::write.xlsx(x = as.data.frame(detected_mirs_per_cond$IP_KO), file = paste0(out,"detected_miR_IP_KO
xlsx::write.xlsx(x = as.data.frame(detected_mirs_per_cond$IP_WT_miR181), file = paste0(out,"detected_mi
xlsx::write.xlsx(x = as.data.frame(detected_mirs_per_cond$IP_KO_miR181), file = paste0(out,"detected_mi
```

4.5.1 Barchart IP_WT/KO topb mirs & IP_mir181_WT/KO

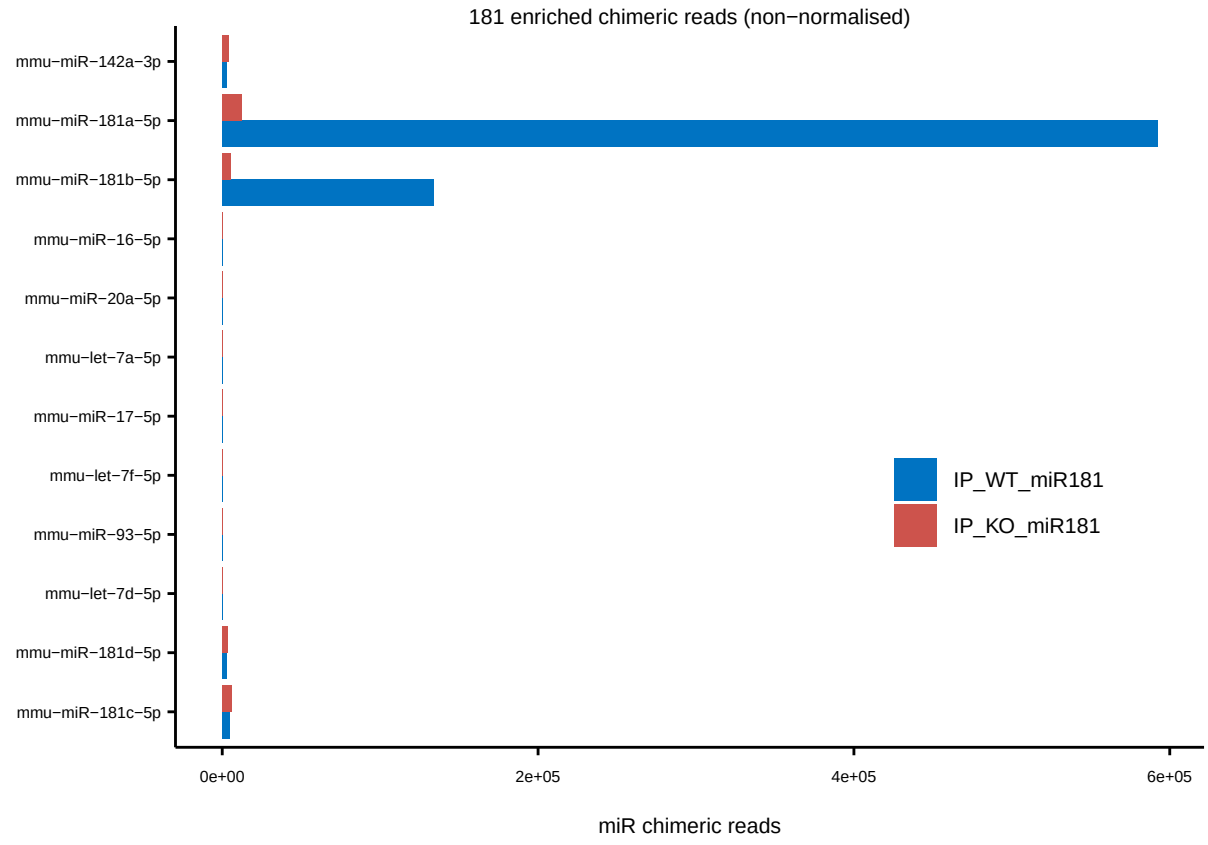
```
# ----------------------------------------
# Barcharts of top miRs per condition
# Figure 1 d & Figure 2 a
# ----------------------------------------


# Barchart of AGO IP
######################

# make one df with all conditions
```

```
conditions_of_samples_list <- rep(condition_names, each =3)
mirs <- pmap(list(x=detected_mirs, y=as.list(sample_names), z= conditions_of_samples_list),
                              function(x,y,z) mutate(x, Sample = y,
                                                     condition = z)) %>%
  map_dfr(~.x)

# get conditions
mirs_ago_wt_ko <- mirs %>% subset(condition %in% c("IP_WT", "IP_KO"))

# calculate relative amount per condition
mirs_ago_wt_ko <- mirs_ago_wt_ko %>%
  mutate( n_total = case_when(condition == "IP_WT" ~ sum(detected_mirs_per_cond$IP_WT$n),
                              condition == "IP_KO" ~ sum(detected_mirs_per_cond$IP_KO$n))) %>%
  group_by(condition, Name) %>%
  mutate(
    n_per_cond_rel = sum(n)/n_total,
    sum = sum(n))

# select top 10 from wt condition
mirs_t10_ago_wt_ko <- mirs_ago_wt_ko %>% subset(Name %in% c(detected_mirs_per_cond_top_10$IP_WT$Name, "
                                                arrange(desc(condition), n_per_cond_rel)



p1 <- ggplot(mirs_t10_ago_wt_ko, aes(x = factor(Name, levels = unique(Name)), y = sum, fill = factor(co
   geom_col( stat="identity",position = "dodge")+
      #scale_x_discrete(guide = guide_axis(angle = 45)) +
  scale_fill_manual(values = c(farbe1, farbe3))+
  xlab("") +
  ylab("miR chimeric reads")+
  coord_flip()

p1 + ggtitle("Ago chimeric reads (non normalised)")
```




```
# Barchart of 181 IP
######################

# get conditions
mirs_181_wt_ko <- mirs %>% subset(condition %in% c("IP_WT_miR181", "IP_KO_miR181"))

# calculate relative amount per condition
mirs_181_wt_ko <- mirs_181_wt_ko %>%
  mutate( n_total = case_when(condition == "IP_WT_miR181" ~ sum(detected_mirs_per_cond$IP_WT_miR181$n),
                              condition == "IP_KO_miR181" ~ sum(detected_mirs_per_cond$IP_KO_miR181$n)))
  group_by(condition, Name) %>%
  mutate(
    n_per_cond_rel = sum(n)/n_total,
    sum = sum(n))

# select top 10 from wt condition
mirs_t10_181_wt_ko <- mirs_181_wt_ko %>% subset(Name %in% c(detected_mirs_per_cond_top_10$IP_WT$Name, "
                                                arrange(desc(condition), n_per_cond_rel)



p3 <- ggplot(mirs_t10_181_wt_ko, aes(x = factor(Name, levels = unique(mirs_t10_ago_wt_ko$Name)), y = su
   geom_col( stat="identity",position = "dodge")+
      #scale_x_discrete(guide = guide_axis(angle = 45)) +
  scale_fill_manual(values = c(farbe1, farbe3))+
```

```
  xlab("") +
  ylab("miR chimeric reads")+
  coord_flip()

p3 + ggtitle("181 enriched chimeric reads (non-normalised)")
```




```
# save for paper
p1 <- p1 + theme_paper()
p3 <- p3 + theme_paper()

ggsave(p1, filename = paste0(out, "Figure1E_Barchart_IP_WT_KO_top_mirs.pdf"), width = unit(6, "cm"), he
ggsave(p3, filename = paste0(out, "Figure1G_Barchart_IP_WT_KO_181enriched_top_mirs.pdf"), width = unit(
```

5 Assign chimeric reads to binding sites

We assign chimeric reads that are in a window of 10nt before the binding site until 10nt after the binding site to the respective binding site.

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

# find overlaps of mirt and AGO bs
idx <- findOverlaps(ago_bs_10,
                    makeGRangesFromDataFrame(mir_crosslinks_per_cond$IP_WT, keep.extra.columns = T))

# make a data frame from the ago bs
```

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

# add mir info to ago bs
ago_bs_chi <- cbind(ago_bs[queryHits(idx),], mir_IP_WT = mir_crosslinks_per_cond$IP_WT[subjectHits(idx)

ago_bs_chi <- ago_bs_chi %>%
  mutate(., n_chimerics = length(mir_IP_WT), .by = c("seqnames", "start", "strand")) %>%
  tidyr::nest(mir_IP_WT)
```

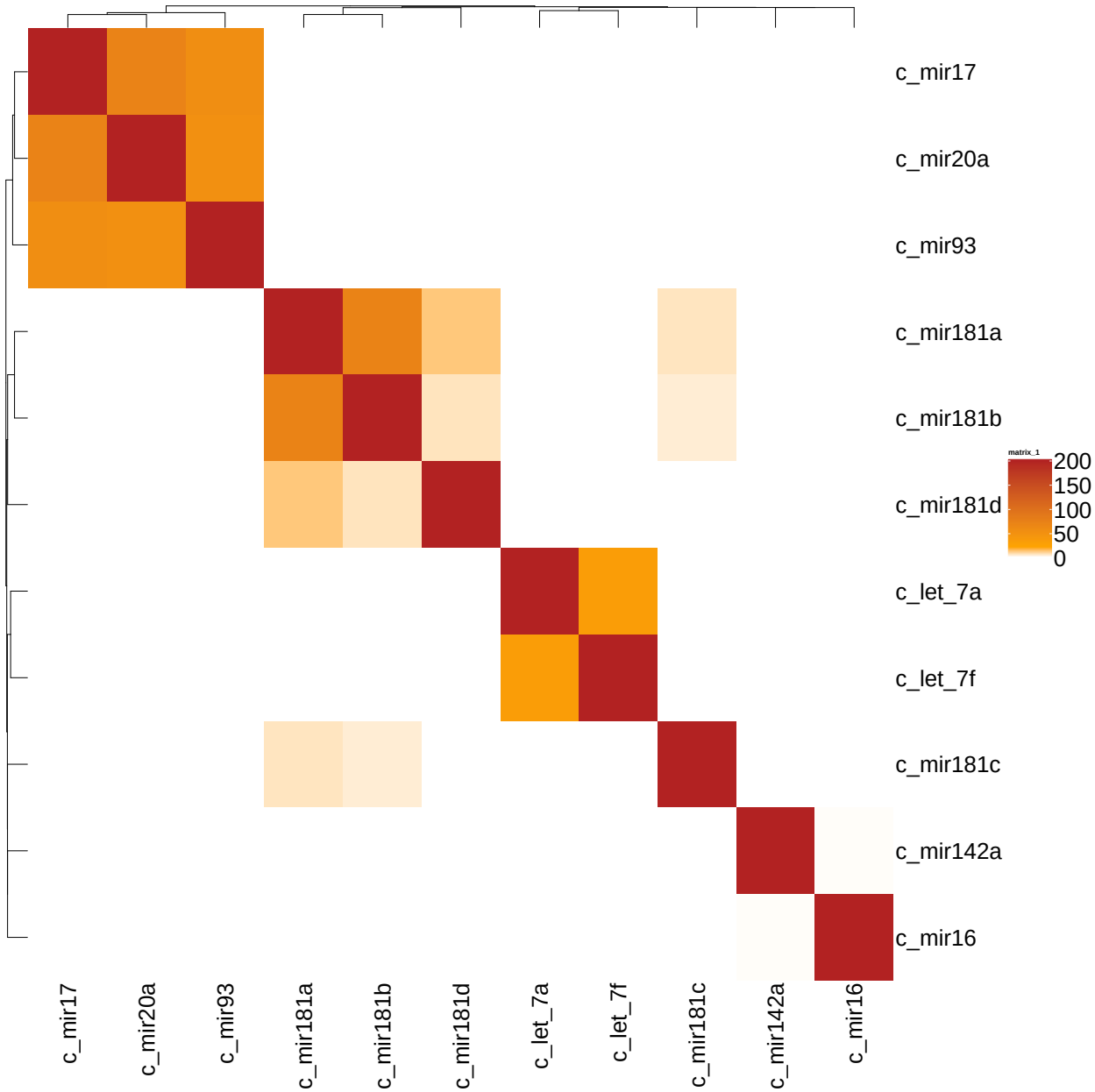
- AGO binding sites with a chimeric read: 7052 , 0.256455

5.1 Enriched sharing of binding sites by two miRs

Ago binding sites can contain more than one miR. Here we look at which miR sharing is enriched. The heatmap shows the r p-values of fisher-tests after bh adjustment.

```
# ----------------------------------------
# Coocruence of multiple miRs in same Ago2 BS
# Figure 1 e
# ----------------------------------------

# get co-occurences of top 10 mirs
t <- ago_bs_chi %>% mutate(n_different_mirs = map(data, ~length(unique(.x$mir_IP_WT))) %>% unlist(.),
                  c_mir181a = map(data, ~ "mmu-miR-181a-5p" %in% .x$mir_IP_WT) %>% unlist(.),
                  c_mir181b = map(data, ~ "mmu-miR-181b-5p" %in% .x$mir_IP_WT) %>% unlist(.),
                  c_mir142a = map(data, ~ "mmu-miR-142a-3p" %in% .x$mir_IP_WT) %>% unlist(.),
                  c_mir16 = map(data, ~ "mmu-miR-16-5p" %in% .x$mir_IP_WT) %>% unlist(.),
                  c_mir20a = map(data, ~ "mmu-miR-20a-5p" %in% .x$mir_IP_WT) %>% unlist(.),
                  c_let_7a = map(data, ~ "mmu-let-7a-5p" %in% .x$mir_IP_WT) %>% unlist(.),
                  c_mir17 = map(data, ~ "mmu-miR-17-5p" %in% .x$mir_IP_WT) %>% unlist(.),
                  c_mir181c = map(data, ~ "mmu-miR-181c-5p" %in% .x$mir_IP_WT) %>% unlist(.),
                  c_mir181d = map(data, ~ "mmu-miR-181d-5p" %in% .x$mir_IP_WT) %>% unlist(.),
                  c_let_7f = map(data, ~ "mmu-let-7f-5p" %in% .x$mir_IP_WT) %>% unlist(.),
                  c_mir93 = map(data, ~ "mmu-miR-93-5p" %in% .x$mir_IP_WT) %>% unlist(.)
                  )


# function for fisher test
fisher_fun <- function(v){
  overlap_m <- eulerr::euler(data.frame(m[,v[[1]]], m[, v[[2]]] ))

  plot(overlap_m , quantities = TRUE, shape = "ellipse")


  v <- overlap_m$original.values
  v <- matrix(c(v[3], v[1], v[2], length(m[,1])), ncol = 2)

  f <- fisher.test(v)
  f$p.value
}

# make matix of top 10 miRs
m <- as.matrix(t[ ,grepl(colnames(t), pattern = "c_")])
```

```r
# calc p-value and adj p-value from pairwise fisher tests
p.values <- combn(x = 1:ncol(m), m = 2, fisher_fun)
p.adj <- p.adjust(p.values)

# make plotable matirx
n <- ncol(m)
mat <- `dimnames<-`(matrix(0,n,n), list(colnames(m), colnames(m)))
mat[lower.tri(mat, diag = F)] <- as.vector(p.adj)

mat <- t(mat) +mat
mat <- -log10(mat)

mat[mat==Inf] <- 200
mat[mat<log10(0.01)] <- 0

# plot with ComplexHeatmap
col_fun = colorRamp2(c(0, 20, 200), c("white",  "orange", "firebrick"))

lgd = list(grid_width = unit(2, "cm"), grid_height= unit(100, "cm"), labels_gp = gpar(fontsize = 30))

h <- Heatmap(mat, col = col_fun,
         row_names_gp = gpar(fontsize = 30),
        row_names_max_width = unit(10, "cm"),
         column_names_gp = gpar(fontsize = 30),
        column_names_max_height = unit(10, "cm"),
        heatmap_legend_param = lgd)
h
```




```
pdf(file = paste0(out, "Figure1F_co-occurence_heatmap.pdf"), width = 11, height = 10)
h
dev.off()
```

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

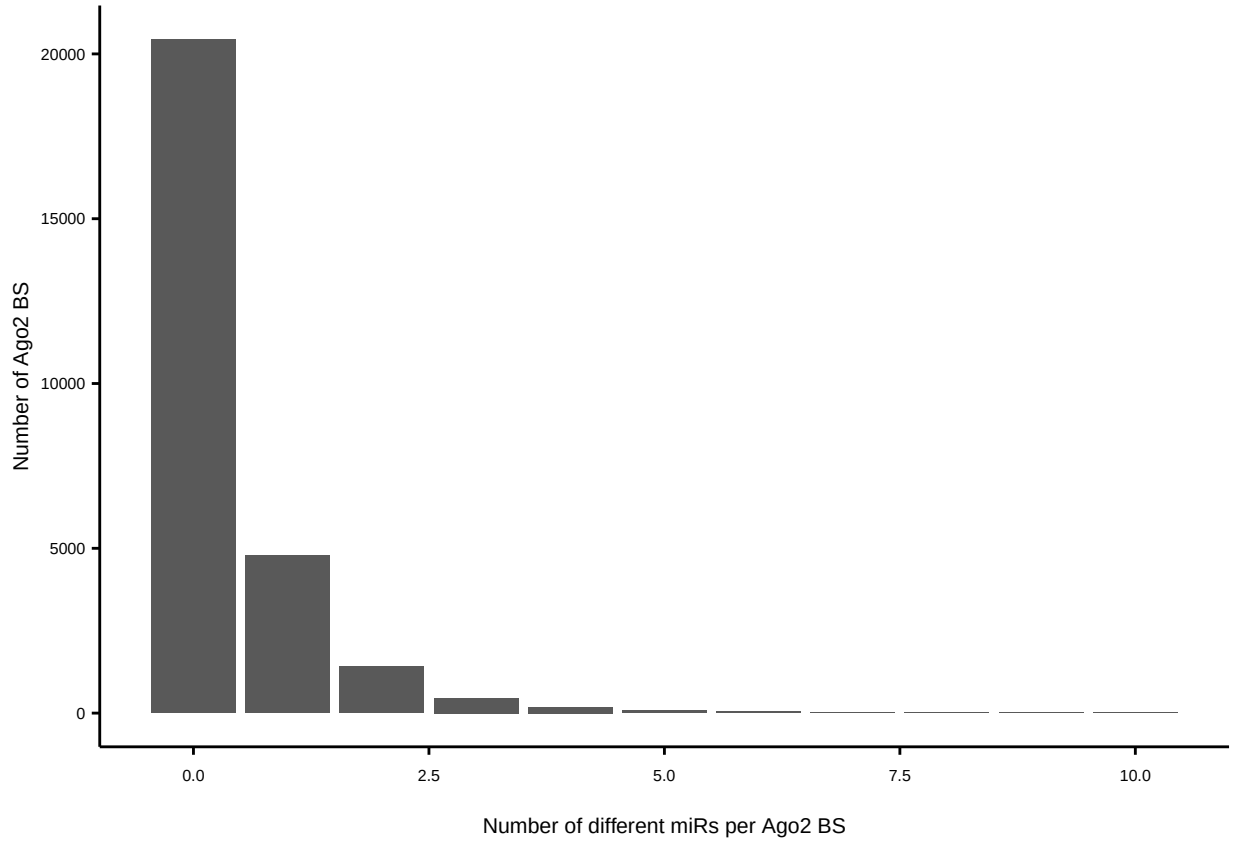
5.2 N different miRs per Ago BS

```
# ---------------------------------------
# Number of different miRs in same Ago2 BS
# Supplementary Figure 1 g
# ---------------------------------------
```

```
t <- table(t$n_different_mirs) %>% as.data.frame()

t10 <- t %>% subset(., as.numeric(Var1) >= 10)
t <- t %>% subset(as.numeric(Var1) <= 10)
t[(as.numeric(t$Var1) == 10),]$Freq <- sum(t10$Freq)
t$Var1 <- as.numeric(t$Var1)

t <- rbind(c(0, nrow(ago_bs) - nrow(ago_bs_chi)), t)

sum(t[3:10,]$Freq)/ sum(t$Freq)
```

```
## [1] 0.08229689
```

```
ggplot(t, aes(x = Var1, y = Freq))+
  geom_col()+
  xlab("Number of different miRs per Ago2 BS")+
  ylab("Number of Ago2 BS")
```



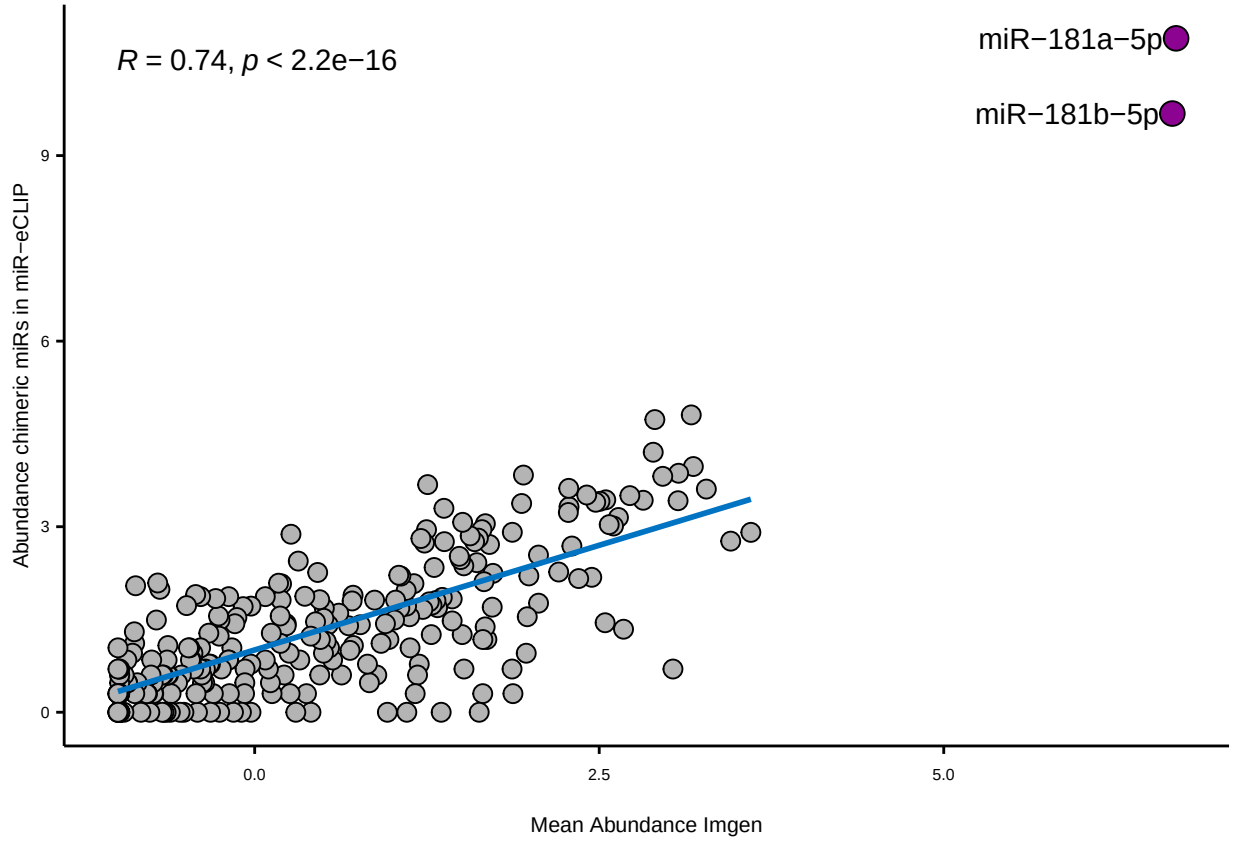

```
ggsave(paste0(out, "Distinct_mirs_per_Ago2_BS.pdf"), width = 5, height = 6, units = "cm")
```

6 Comparison to mir181 quantification from imgen

Here we compare detected miR amounts to the publicity available quantification from Immgen.

```
# ----------------------------------------
# Comparison of miR numbers to PCR abundace from immgen
```

```r
# Figure 1 c
# ----------------------------------------

comp_df <- table(mir_crosslinks_per_cond$IP_WT$Name) %>% as.data.frame(.) %>%
  rowwise(.)%>%
  mutate(Var1 = substr(Var1, 5, nchar(as.character(Var1))))

mir_imgen <- mir_imgen %>% full_join(comp_df, by = c(ID_REF = "Var1"))

p4 <- ggplot(mir_imgen, aes(x = log10(mean), y = log10(Freq)))+
  geom_point(fill="#b4b4b4", colour="black", size=3, shape=21)+ #changed by nikita
  geom_point(data=mir_imgen[mir_imgen$ID_REF == "miR-181a-5p",],aes(x = log(mean), y = log(Freq)), fill=
  geom_point(data=mir_imgen[mir_imgen$ID_REF == "miR-181b-5p",],aes(x = log(mean), y = log(Freq)), fill=
  geom_text_repel(data=mir_imgen[mir_imgen$ID_REF %in% c("miR-181a-5p", "miR-181b-5p"),], aes(x = log(m
  geom_smooth(method =lm,se=F, colour=farbe1)+
  stat_cor()+
  xlab("Mean Abundance Imgen")+
  ylab("Abundance chimeric miRs in miR-eCLIP")

p4
```




```r
p4 <- p4 + theme_paper()

ggsave(p4, filename = paste0(out, "Figure1c_Comparison_miR_expression_Imgen_nv.pdf"), width = unit(6, "
```

7 Save files

```
saveRDS(mir_crosslinks_per_cond, file = paste0(out, "mir_chimeric_crosslinks.rds"))
```

8 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] ggrepel_0.9.3                     ggpubr_0.6.0
##  [3] circlize_0.4.15                   BSgenome.Mmusculus.UCSC.mm10_1.4.3
##  [5] BSgenome_1.66.3                   Biostrings_2.66.0
##  [7] XVector_0.38.0                    ComplexHeatmap_2.14.0
##  [9] kableExtra_1.3.4                  rtracklayer_1.58.0
## [11] GenomicRanges_1.50.2              GenomeInfoDb_1.34.9
## [13] IRanges_2.32.0                    S4Vectors_0.36.2
## [15] BiocGenerics_0.44.0               purrr_1.0.1
## [17] dplyr_1.1.2                       ggplot2_3.4.2
## [19] knitr_1.43
##
## loaded via a namespace (and not attached):
##  [1] colorspace_2.1-0            ggsignif_0.6.4
##  [3] rjson_0.2.21                rprojroot_2.0.3
##  [5] GlobalOptions_0.1.2         clue_0.3-64
##  [7] rstudioapi_0.15.0           farver_2.1.1
##  [9] fansi_1.0.4                 xml2_1.3.5
## [11] splines_4.2.2               codetools_0.2-19
## [13] doParallel_1.0.17           polyclip_1.10-4
## [15] Rsamtools_2.14.0            Cairo_1.6-0
## [17] rJava_1.0-6                 broom_1.0.5
## [19] cluster_2.1.4               png_0.1-8
## [21] compiler_4.2.2              httr_1.4.6
## [23] backports_1.4.1             Matrix_1.5-4.1
## [25] fastmap_1.1.1               cli_3.6.1
## [27] htmltools_0.5.5             tools_4.2.2
## [29] gtable_0.3.3                glue_1.6.2
## [31] GenomeInfoDbData_1.2.9      Rcpp_1.0.11
## [33] carData_3.0-5               Biobase_2.58.0
```

```
## [35] eulerr_7.0.0                vctrs_0.6.3
## [37] nlme_3.1-162                svglite_2.1.1
## [39] iterators_1.0.14            xfun_0.39
## [41] polylabelr_0.2.0            stringr_1.5.0
## [43] xlsxjars_0.6.1              rvest_1.0.3
## [45] lifecycle_1.0.3             restfulr_0.0.15
## [47] rstatix_0.7.2               XML_3.99-0.14
## [49] xlsx_0.6.5                  zlibbioc_1.44.0
## [51] scales_1.2.1                ragg_1.2.5
## [53] MatrixGenerics_1.10.0       parallel_4.2.2
## [55] SummarizedExperiment_1.28.0 RColorBrewer_1.1-3
## [57] yaml_2.3.7                  stringi_1.7.12
## [59] highr_0.10                  BiocIO_1.8.0
## [61] foreach_1.5.2               BiocParallel_1.32.6
## [63] shape_1.4.6                 rlang_1.1.1
## [65] pkgconfig_2.0.3             systemfonts_1.0.4
## [67] matrixStats_1.0.0           bitops_1.0-7
## [69] evaluate_0.21               lattice_0.21-8
## [71] GenomicAlignments_1.34.1    labeling_0.4.2
## [73] tidyselect_1.2.0            here_1.0.1
## [75] magrittr_2.0.3              R6_2.5.1
## [77] magick_2.7.4                generics_0.1.3
## [79] DelayedArray_0.24.0         pillar_1.9.0
## [81] withr_2.5.0                 mgcv_1.8-42
## [83] abind_1.4-5                 RCurl_1.98-1.12
## [85] tibble_3.2.1                crayon_1.5.2
## [87] car_3.1-2                   utf8_1.2.3
## [89] rmarkdown_2.23              GetoptLong_1.0.5
## [91] digest_0.6.33               webshot_0.5.5
## [93] tidyr_1.3.0                 textshaping_0.3.6
## [95] munsell_0.5.0               viridisLite_0.4.2
```