RNAduplex on all binding sites, (236nt window)

Melina Klostermann

22 September, 2023

Contents

1 Libraries and settings

```
# ----------------------------------------
# libraries
# ----------------------------------------
library(tidyverse)
library(GenomicRanges)
library(colorspace)
library(gghalves)
library(BSgenome.Mmusculus.UCSC.mm10)
library(Biostrings)
library(ComplexHeatmap)
library(ggpubr)
library(circlize)


here <- here::here()

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

out <- paste0(here,"/Figure6/01_RNAduplex/")
source(paste0(here,"/Supporting_scripts/themes/theme_paper.R"))
```

```
source(paste0(here,"/Supporting_scripts/themes/CustomThemes.R"))

set.seed(2)
```

2 What was done?

- run RNAduplex on a region of 10nt before until 30nt after all 6mer seeds in the expressed transcriptome
- select bound seeds (in 200nt after a mir181 enriched binding site)
- look at mir181 structure in duplexes with bound seeds
- cluster mir181 structures (different number of clusters tested)
- look at nucleotide contribution in total an per cluster (also combined nucleotide contribution of 2 or 3 bound in a row)
- check binding site strength and minimum free energy per cluster

3 Files

```
# ----------------------------------------
# MREs
# ----------------------------------------

mir181_bs <- readRDS(paste0(here, "/Figure5/01_Seed_motifes/mir181_bs_with_seeds_transcripts.rds"))

mir181_enriched_set <- mir181_bs %>%
  as.data.frame(.) %>%
  subset(set %in% c("ago_bs_mir181_chi&mir181_enriched", "mir181_enriched"))

# transcript seqeuces
transcript_fasta <- readDNAStringSet("/Users/melinaklostermann/Documents/projects/anno/gencodevM23/genco

# annotation
anno <- readRDS(paste0(here,"/Supporting_scripts/annotation_preprocessing/annotation.rds"))

# expressed_genes
expressed_genes <- readRDS(paste0(here, "/Supporting_scripts/TPMs-RNAseq/expressed_genes.rds"))

# seeds
seeds <- read.csv(paste0(here,"/Figure3/01_Ribosome_profiling_pipeline/RPF_masterframe.csv"))

# Ribofootprint
rfp <- read.csv(paste0(here,"/Figure3/01_Ribosome_profiling_pipeline/RPF_masterframe.csv") )
rfp <- rfp %>% mutate(Gene = sub("\\..*", "", Gene))
```

4 Run RNAplfold on all 6mers

4.1 Get transcript sequences

```
###################
# Get sequences of mature transcripts
##################

# expressed transcripts ( = transcripts with any crosslinks)
```

```
annotation_transcripts <- anno[anno$type == "transcript"]
expressed_transcripts <- annotation_transcripts[annotation_transcripts$geneID %in% expressed_genes]


# transcript seqeunces
transcript_anno_meta <- names(transcript_fasta)
transcript_anno_meta <- data.frame(all = transcript_anno_meta) %>%
  tidyr::separate(., col = all,
                  into = c("transcript_id", "gene_id", "a", "b", "isoform_name", "gene_name", "entrez_g

names(transcript_fasta) <- sub("\\..*", "", transcript_anno_meta$transcript_id)

# get transcript_id and transcript lengths from fasta names
transcript_fasta_df <- data.frame(tx_name = names(transcript_fasta), width = width(transcript_fasta))
```

4.2 all binding sites

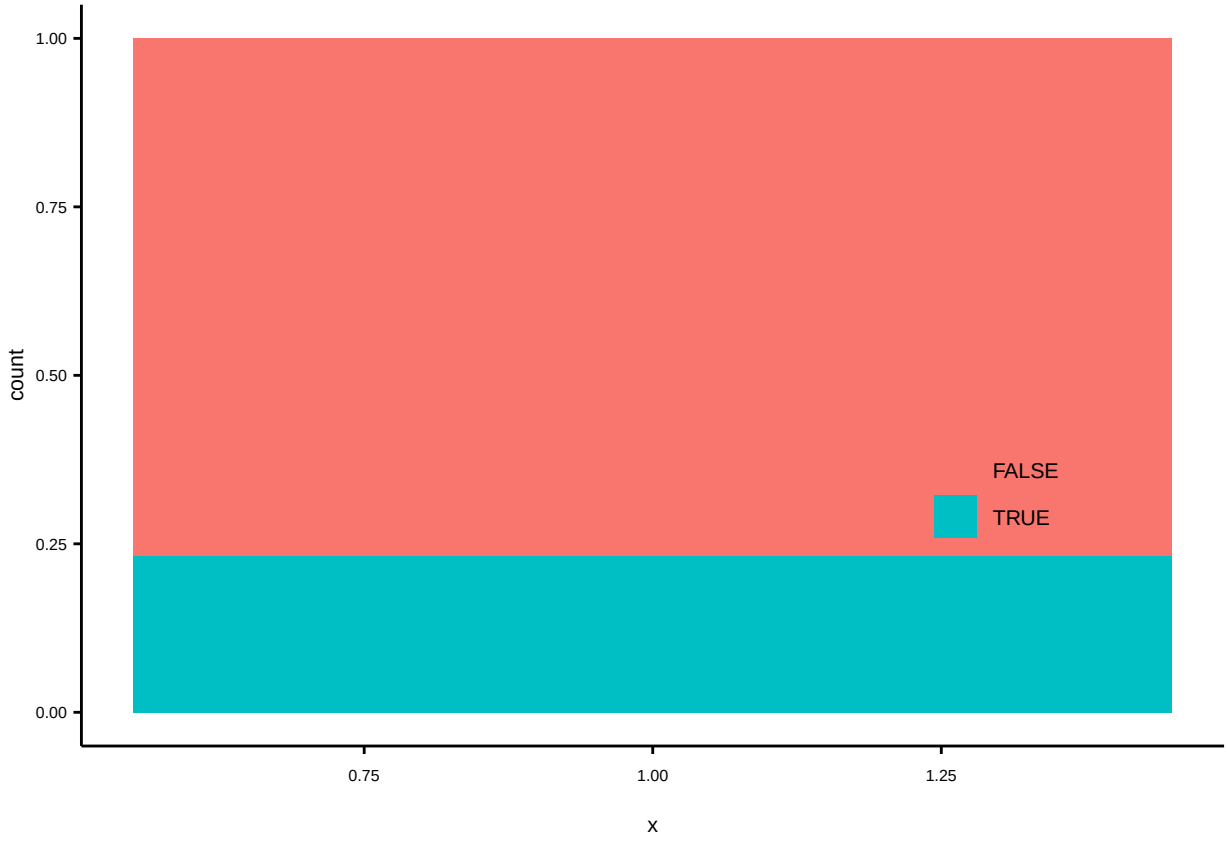
```
# make window around mir181 binding sites
w <- 237
mir181_enriched_set_237nt <- mir181_enriched_set %>%
  left_join(transcript_fasta_df, by= c(seqnames = "tx_name"), suffix = c(".bs", ".tx")) %>%
  mutate(end = end + 200, start = start -30, strand = "*") %>%
  dplyr::filter((end <  width.tx)  & (start > 0)) %>%
  makeGRangesFromDataFrame(., keep.extra.columns = T) %>%
  unique(.)

mir181_enriched_set_237nt <- mir181_enriched_set_237nt[width(mir181_enriched_set_237nt)==w]

mir181_enriched_set_237nt_seqs <- Biostrings::getSeq(x = transcript_fasta, names = mir181_enriched_set_

NROW(mir181_enriched_set_237nt)
```

```
## [1] 4073
```

```
# oneline fasta
writeXStringSet(mir181_enriched_set_237nt_seqs, filepath = paste0(out,"mir181_enriched_set_237nt.fasta"

# specific column for 6mer seeds
seed_from_237nt <- as.data.frame(mir181_enriched_set_237nt) %>%
  mutate(end = end - 200, start = start +30) %>%
  unnest(all_seeds_200down)

seed_from_237nt <- seed_from_237nt %>%
    subset(., ((.$Seeds_200down.type %in% c("seed_6mer", "seed_6mer_wobble")) | is.na(.$Seeds_200down.ty
  group_by(mir181BS_ID) %>%
  arrange(Seeds_200down.start, .by_group = T) %>%
  dplyr::slice(1) %>%
  ungroup() %>%
  mutate(Seeds_200down.type = case_when(is.na(Seeds_200down.type) ~ "no_seed",
                                        T ~ Seeds_200down.type))
```

4.3 Number of binding sites with seed motif

```
p <- ggplot(seed_from_237nt, aes(x = 1, fill = (Seeds_200down.type == "seed_6mer") )) +
  geom_bar( position = "fill")+
  theme_paper()

p
```




```
t <- table(seed_from_237nt$Seeds_200down.type == "seed_6mer")
t
```

```
##
## FALSE  TRUE
##  2898   873
```

```
t / sum(t)
```

```
##
##     FALSE      TRUE
## 0.7684964 0.2315036
```

```
ggsave(p, filename = paste0(out, "SuppFigure6A_MRE_with_seed_bar.pdf"), width = 6, height = 6, units =
```

4.4 RNAduplex output

```
# --------------------
# Read in RNAduplex output and clean
# --------------------
```

```
struct <- read_table(paste0(out,"/mir181_enriched_set_237nt.struct"), col_names = c("seq", "mir", "stru
```

```
struct <- struct %>%
  rowwise(.) %>%
  mutate(struct_mir = str_split_1(structure, pattern = "&")[2],
         struct_target = str_split_1(structure, pattern = "&")[1],
         start_mir = str_split_1(position_mir, pattern = ",")[1] %>% as.numeric(.),
         end_mir = str_split_1(position_mir, pattern = ",")[2] %>% as.numeric(.),
         start_target = str_split_1(position_seq, pattern = ",")[1] %>% as.numeric(.),
         end_target = str_split_1(position_seq, pattern = ",")[2] %>% as.numeric(.),
         min_free_energy = gsub("[()]", "", min_free_energy) %>% as.numeric(.),
         norm_free_energy = min_free_energy / (nchar(structure)-1),
         # the last bound position in the target = the position that is bound by the beginning of the m
         end_target_bound = end_target - nchar(str_split_1(rev(struct_target), pattern = "[(]")[1]))

struct <- struct %>%
  mutate(struct_bound_mir_full = paste0(
    paste0( rep(".", start_mir), collapse = ""),
    struct_mir,
    paste0(rep(".", (23 - end_mir)), collapse = ""),
    collapse = ""))

struct$mir181BS_ID <- mir181_enriched_set_237nt$mir181BS_ID

head(struct)
```

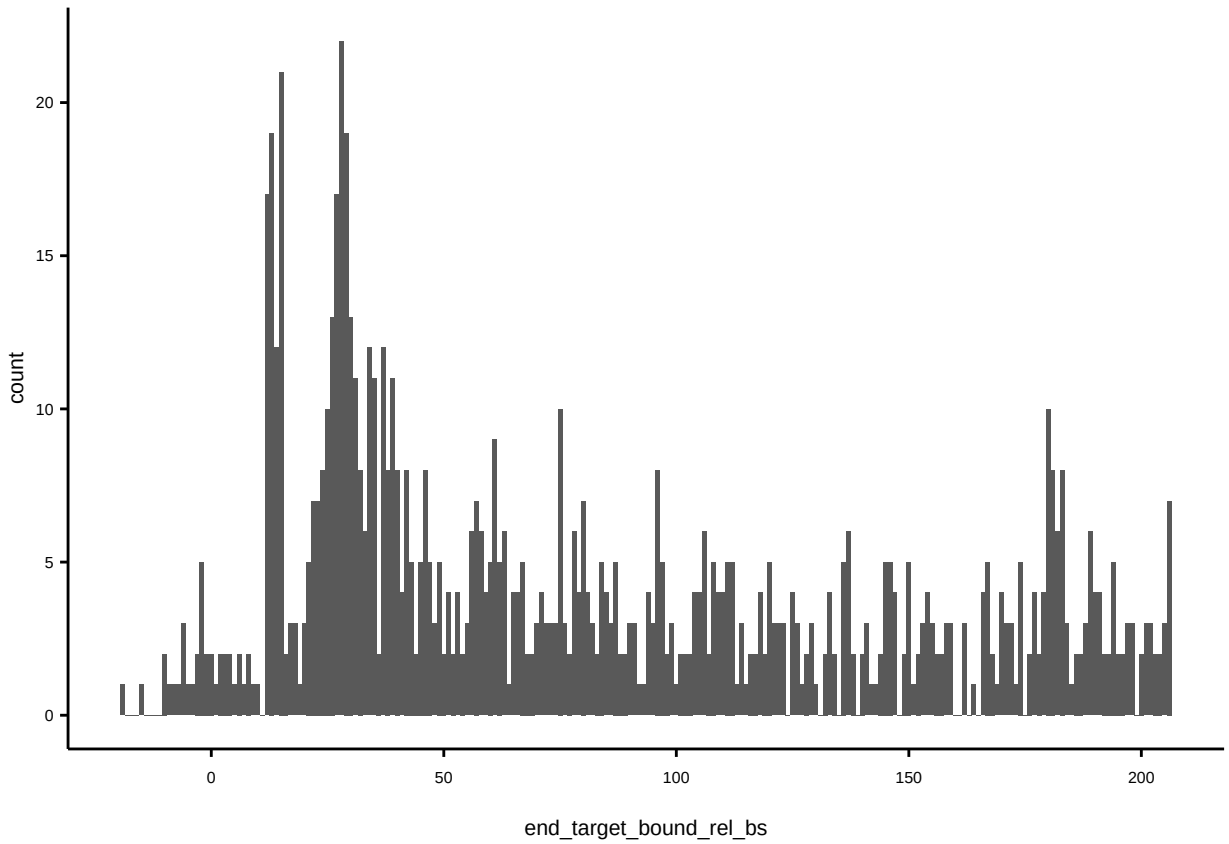
```
## # A tibble: 6 x 17
## # Rowwise:
##   seq   mir   structure          position_seq x     position_mir min_free_energy
##   <chr> <chr> <chr>              <chr>        <chr> <chr>                  <dbl>
## 1 >seq  >mir  .((((((......(((.~ 2,23         :     9,23                   -12.2
## 2 >seq  >mir  .((((((...(((((..~ 41,68        :     2,23                   -13.5
## 3 >seq  >mir  .(((.(((((((.....~ 41,70        :     1,23                   -21.1
## 4 >seq  >mir  .(((((((((((.(((~ 200,218      :     1,20                   -12.6
## 5 >seq  >mir  .(((((.(((((.((.(~ 13,34        :     2,23                   -13.1
## 6 >seq  >mir  .(((..((.((((((.&~ 70,85        :     10,23                  -13.1
## # i 10 more variables: struct_mir <chr>, struct_target <chr>, start_mir <dbl>,
## #   end_mir <dbl>, start_target <dbl>, end_target <dbl>,
## #   norm_free_energy <dbl>, end_target_bound <dbl>,
## #   struct_bound_mir_full <chr>, mir181BS_ID <int>
```

```
# --------------------
# make structur matrix of mir
# --------------------
struct_bound_mir_mat <- data.frame(s = struct$struct_bound_mir_full)
struct_bound_mir_mat <- struct_bound_mir_mat %>% separate(., s, as.character(1:25), sep = "")
struct_bound_mir_mat <- as.matrix(struct_bound_mir_mat)
struct_bound_mir_mat <- struct_bound_mir_mat[,-1]
n <- ncol(struct_bound_mir_mat)

struct_bound_mir_mat[struct_bound_mir_mat == ")"] = 1
struct_bound_mir_mat[struct_bound_mir_mat == "."] = 0
struct_bound_mir_mat[struct_bound_mir_mat == ""] = NA
```

```
struct_bound_mir_mat <- as.numeric(struct_bound_mir_mat) %>% matrix(., ncol = n)

# --------------------
# make data frame with extra info
# --------------------
struct_bound_mir_df <- as.data.frame(struct_bound_mir_mat)

st <- struct %>%
  as.data.frame(.) %>%
  dplyr::select( min_free_energy, start_target, end_target, end_target_bound, norm_free_energy, mir181BS

struct_bound_mir_df <- cbind(struct_bound_mir_df, struct)

# info from mir binding sites
s <- seed_from_237nt %>%
  dplyr::select( seqnames, scoreMax, region, resBs.log2FoldChange, Seeds_200down.type, Seeds_200down.sta

struct_bound_mir_df <- left_join(struct_bound_mir_df, s, by = "mir181BS_ID")
```

4.5 Duplex start in relation to 6mer start

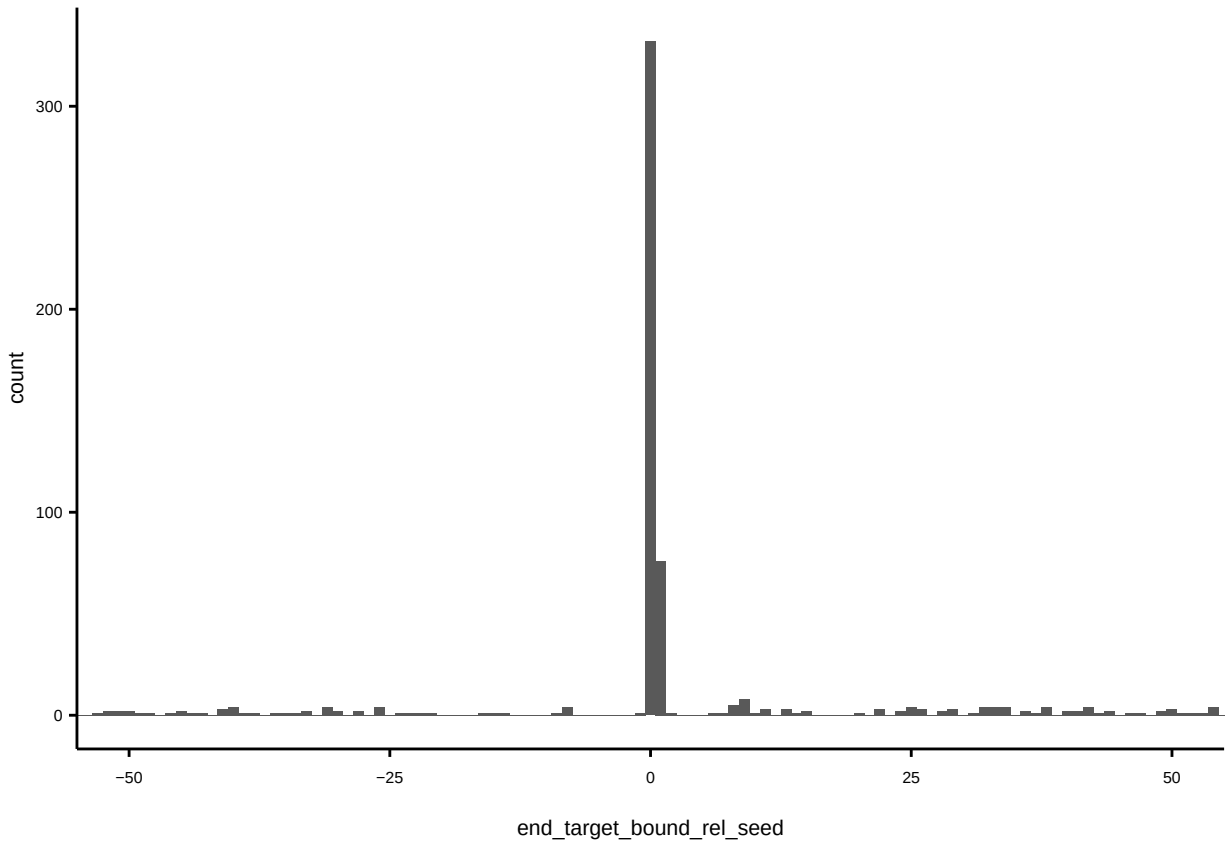
```
struct_bound_mir_df$end_target_bound_rel_bs <- struct_bound_mir_df$end_target_bound -30
struct_bound_mir_df$end_target_bound_rel_seed <- struct_bound_mir_df$end_target_bound_rel_bs - struct_bo

struct_bound_mir_df_6mer <- struct_bound_mir_df %>% subset(Seeds_200down.type == "seed_6mer")

ggplot(struct_bound_mir_df_6mer, aes(x = end_target_bound_rel_bs ))+
  geom_histogram( binwidth = 1)+
  theme_paper()
```




```
nrow(struct_bound_mir_df_6mer)
```

```
## [1] 873
```

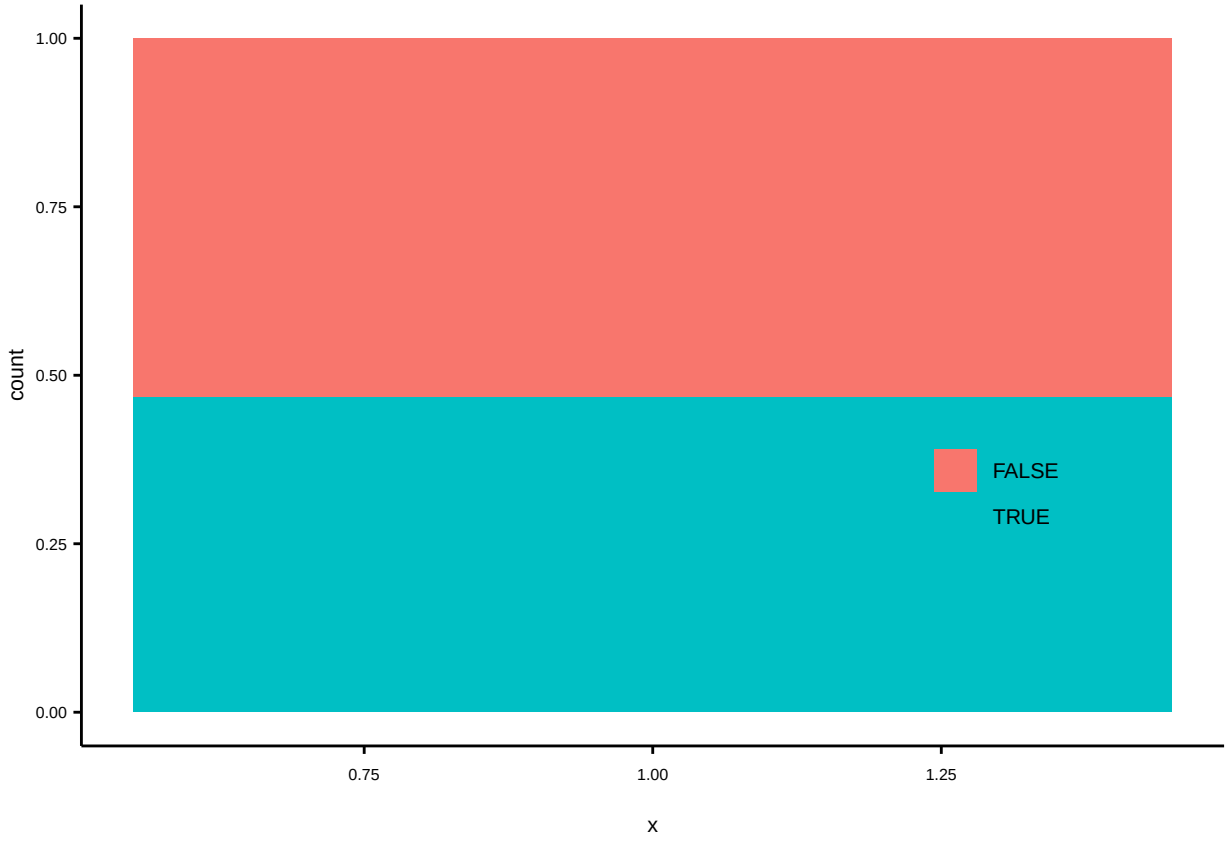
```
p1 <- ggplot(struct_bound_mir_df_6mer, aes(x = end_target_bound_rel_seed ))+
  geom_histogram( binwidth = 1)+
  theme_paper()+
  coord_cartesian(xlim=c(-50,50))

p1
```




```
ggsave(p1, filename = paste0(out, "SuppFigure6C_duplex_start_position.pdf"), width = 6, height = 4, uni
```

4.6 Duplexes that use correct 6mer

```
struct_bound_mir_df_6mer <- struct_bound_mir_df_6mer %>%
  rowwise(.) %>%
  mutate(correct_duplex_end = end_target_bound_rel_seed %in% c(0,1),
         paired_6mer = all(across(V3:V8) == 1),
         canonical_duplex = all(c(correct_duplex_end, paired_6mer)))


p2 <- ggplot(struct_bound_mir_df_6mer, aes(x = 1, fill = canonical_duplex)) +
  geom_bar( position = "fill")+
  theme_paper()

p2
```




```
t <- table(struct_bound_mir_df_6mer$canonical_duplex)
t
```

```
##
## FALSE  TRUE
##   465   408
```

```
t/sum(t)
```

```
##
##     FALSE      TRUE
## 0.532646 0.467354
```

```
ggsave(p2, filename = paste0(out, "SuppFigure6C_canonical_duplex_seeds_bar.pdf"), width = 6, height = 6
```

5 Heatmap of canonical seed pairing

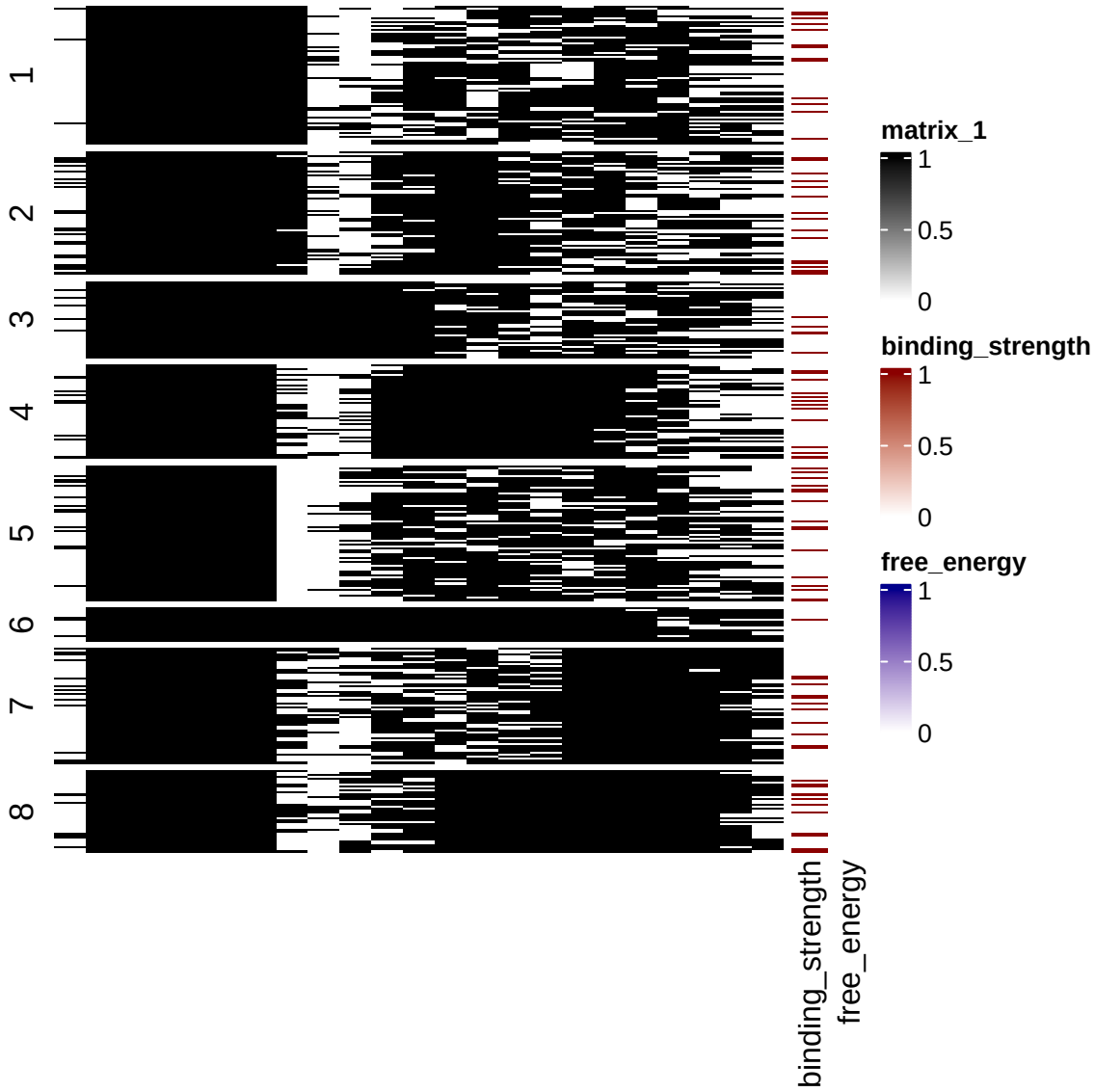
```
# get mir181 pairing of canonical bound RNAs
can_seed <- struct_bound_mir_df_6mer %>% subset(canonical_duplex == T)
mat_can_seed <- can_seed %>% dplyr::select(V2:V24) %>%
  as.matrix()

# color for heatmap
col_fun <- colorRamp2(colors = c("white", "black"), breaks = c(0, 1))

# make a matix with conseqtive 1 added up
mat_can_seed_cons <- apply(mat_can_seed, 1, function(x){
  r = rle(x)
```

```
  r2 = r$lengths*r$values
  r3 = rep(r2,r$lengths)
  return(r3)
})

mat_can_seed_cons <- t(mat_can_seed_cons)
mat_can_seed_cons[1:10,]
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
##  [1,]    0    6    6    6    6    6    6    0    0     0     0     7     7
##  [2,]    0    6    6    6    6    6    6    0    0     0    10    10    10
##  [3,]    9    9    9    9    9    9    9    9    9     0     3     3     3
##  [4,]    0    6    6    6    6    6    6    0    0     5     5     5     5
##  [5,]    0   11   11   11   11   11   11   11   11    11    11    11     0
##  [6,]    0    7    7    7    7    7    7    7    0     0    12    12    12
##  [7,]    0    6    6    6    6    6    6    0    0     0     3     3     3
##  [8,]    0    6    6    6    6    6    6    0    0    12    12    12    12
##  [9,]    0    6    6    6    6    6    6    0    0     0     0    10    10
## [10,]    0    8    8    8    8    8    8    8    8     0     0     0     2
##       [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23]
##  [1,]     7     7     7     7     7     0     3     3     3     0
##  [2,]    10    10    10    10    10    10    10     0     0     0
##  [3,]     0     9     9     9     9     9     9     9     9     9
##  [4,]     5     0     0     6     6     6     6     6     6     0
##  [5,]     0     0     0     5     5     5     5     5     0     0
##  [6,]    12    12    12    12    12    12    12    12    12     0
##  [7,]     0     2     2     0     5     5     5     5     5     0
##  [8,]    12    12    12    12    12    12    12    12     0     0
##  [9,]    10    10    10    10    10    10    10    10     0     0
## [10,]     2     0     0     7     7     7     7     7     7     7
```

```
set.seed(2)
k <- kmeans(mat_can_seed_cons, centers = 8)
can_seed$kmeans <- k$cluster

# side annotations
col1 = colorRamp2(colors = c("white", "darkred"), breaks = c(0, 1))
col2 = colorRamp2( c("white", "darkblue"), breaks = c(0, 1))


binding_top_20 <- log10(can_seed$scoreMax)
binding_top_20[binding_top_20 < quantile(binding_top_20, probs = seq(0, 1, 0.2))["80%"]] <- 0
binding_top_20[binding_top_20 >= quantile(binding_top_20, probs = seq(0, 1, 0.2))["80%"]] <- 1

free_energy_top_20 <- can_seed$norm_free_energy
free_energy_top_20[free_energy_top_20  <= quantile(free_energy_top_20, probs = seq(0, 1, 0.2))["20%"]] <
free_energy_top_20[free_energy_top_20  > quantile(free_energy_top_20, probs = seq(0, 1, 0.2))["20%"]] <-

ra <- rowAnnotation(binding_strength = binding_top_20,
                    free_energy = free_energy_top_20,
                    col = list(binding_strength = col1,
                               free_energy = col2 ))
```

```
# plot heatmaps

Heatmap(mat_can_seed_cons, cluster_rows = F, cluster_columns = F, col = col_fun, split = k$cluster, rig
```



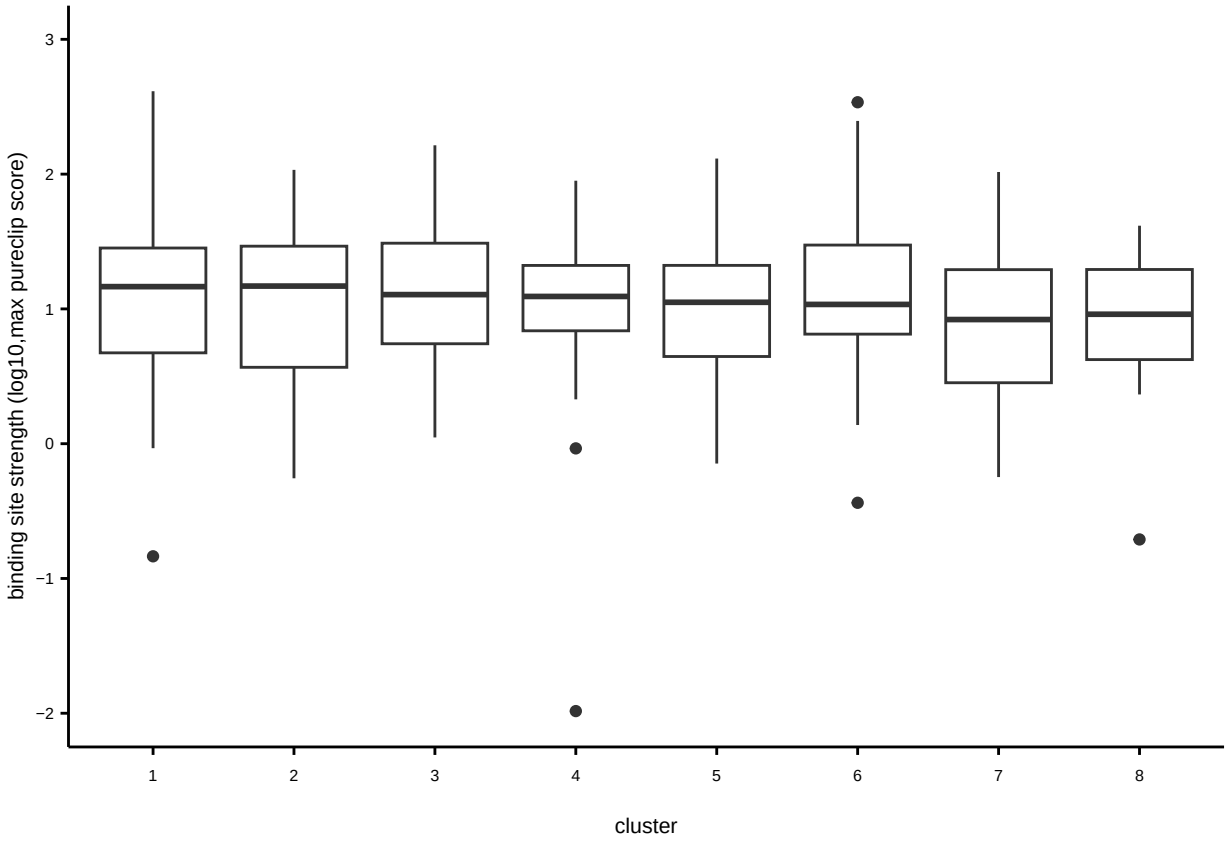

```
pdf(file = paste0(out,"Figure6C_Heatmap_with_seed.pdf"))
Heatmap(mat_can_seed_cons, cluster_rows = F, cluster_columns = F, col = col_fun, split = k$cluster, rig
dev.off()
```

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

5.1 Bindingsite strength per cluster

```
names_reordered_clusters <- data.frame( cluster = c(1,2,3,4,5,6,7,8), reordered_cluster  = c(5,6,7,2,1,

can_seed <- left_join(can_seed, names_reordered_clusters, by = c(kmeans = "cluster") )

ggplot(can_seed, aes(as.character(reordered_cluster), log10(scoreMax)))+
```

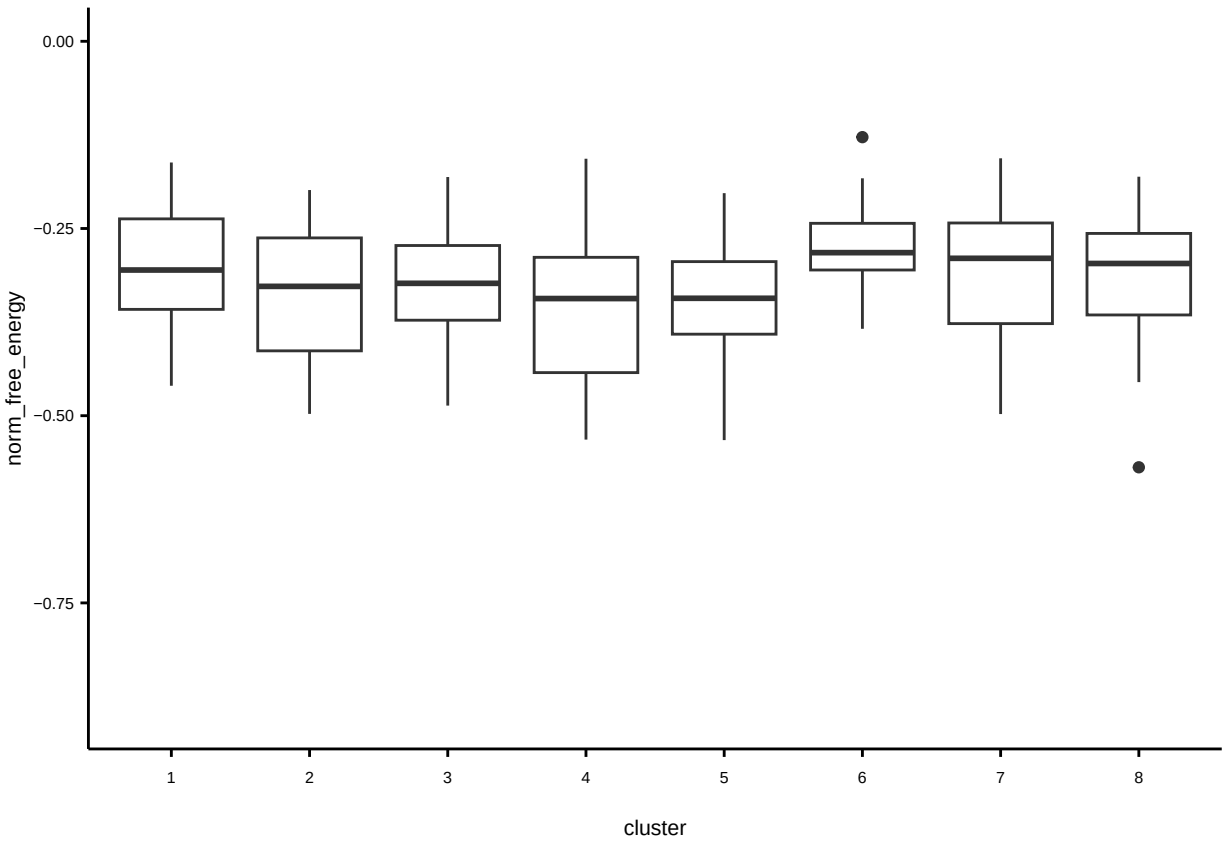
```
  geom_boxplot()+
  coord_cartesian(ylim = c(-2,3))+
  theme_paper()+
  xlab("cluster")+
  ylab("binding site strength (log10,max pureclip score)")
```




```
ggsave( filename = paste0(out, "SuppFigureH_pureclip_score_per_cluster_can.pdf"), width = 6, height = 5
```

5.2 Free energy per cluster

```
ggplot(can_seed, aes(as.character(kmeans), norm_free_energy))+
  geom_boxplot()+
  theme_paper()+
  coord_cartesian(ylim= c(-0.9,0))+
  xlab("cluster")
```




```
ggsave( filename = paste0(out, "FigureS6E_min_free_energy_per_cluster_can.pdf"), width = 6, height = 5,
```

5.3 N per cluster

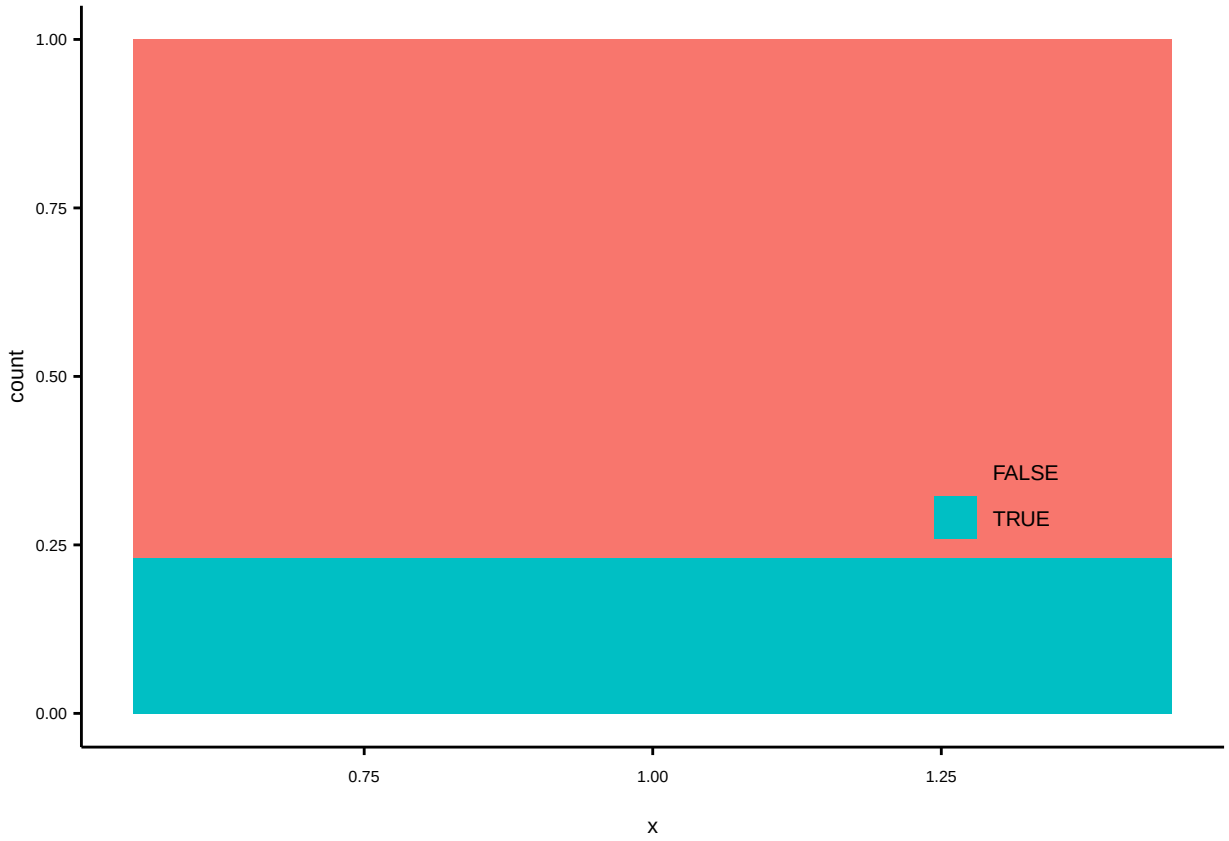
```
table(can_seed$reordered_cluster)
```

```
##
##  1  2  3  4  5  6  7  8
## 69 48 42 59 71 63 39 17
```

6 Heatmap of non-canonical seed pairing

```
noncan_seed <- struct_bound_mir_df %>% subset(Seeds_200down.type != "seed_6mer")

noncan_seed <- noncan_seed %>%
  rowwise(.) %>%
  mutate(struct_mir_8mer = substr(struct_bound_mir_full,2,9),
         n_pairing_mir_8mer = str_count(struct_mir_8mer, "[)]"),
         no_8mer_pairing = n_pairing_mir_8mer == 0,
         temp = list(str_split_1(struct_mir_8mer, "[)]")),
         n_non_pairing_3p_mir_8mer = nchar(temp[[length(temp)]]),
         temp = NULL
        )


p2 <- ggplot(noncan_seed, aes(x = 1, fill = no_8mer_pairing)) +
```

```
  geom_bar( position = "fill")+
  theme_paper()

p2
```




```
t <- table(noncan_seed$no_8mer_pairing)
t
```

```
##
## FALSE  TRUE
##  2229   669
```

```
t/sum(t)
```

```
##
##     FALSE      TRUE
## 0.7691511 0.2308489
```

```
ggsave(p2, filename = paste0(out, "SuppFigure6B_no_seed_seed_binding_bar.pdf"), width = 6, height = 6, u
```

6.1 Heamap with no binding in first 7 nt

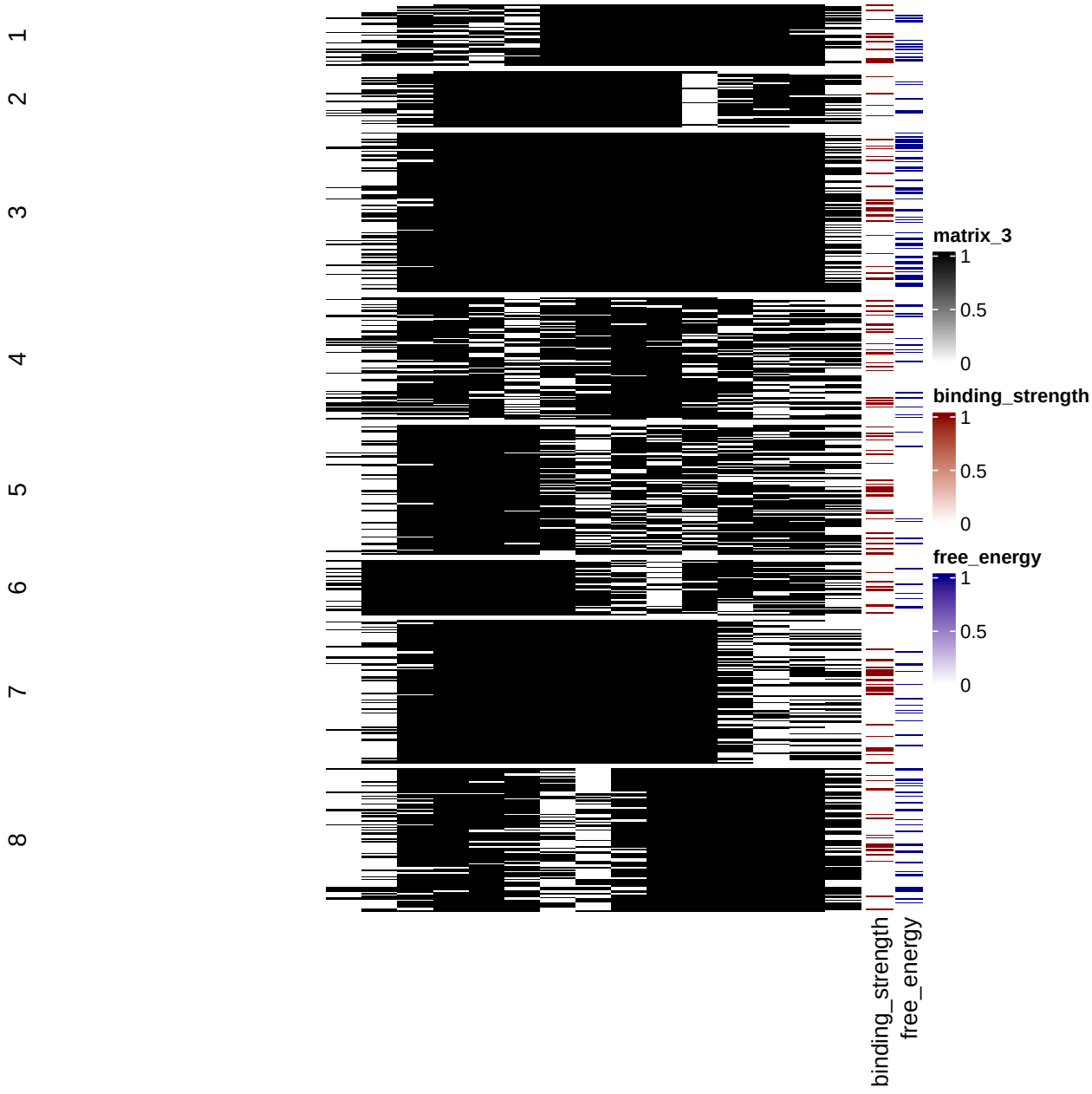
```
# get mir181 pairing of no bound RNAs
mat_noncan_seed_1 <- noncan_seed %>%
  subset(no_8mer_pairing) %>%
  dplyr::select(V2:V24) %>%
  as.matrix()
```

```r
# color for heatmap
col_fun <- colorRamp2(colors = c("white", "black"), breaks = c(0, 1))

# make a matix with conseqtive 1 added up
mat_noncan_seed_cons_1 <- apply(mat_noncan_seed_1, 1, function(x){
  r = rle(x)
  r2 = r$lengths*r$values
  r3 = rep(r2,r$lengths)
  return(r3)
})

mat_noncan_seed_cons_1 <- t(mat_noncan_seed_cons_1)
mat_noncan_seed_cons_1[1:10,]
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
##  [1,]    0    0    0    0    0    0    0    0    0    13    13    13    13
##  [2,]    0    0    0    0    0    0    0    0    0     0     6     6     6
##  [3,]    0    0    0    0    0    0    0    0    0     0    12    12    12
##  [4,]    0    0    0    0    0    0    0    0    7     7     7     7     7
##  [5,]    0    0    0    0    0    0    0    0    0     0     0     8     8
##  [6,]    0    0    0    0    0    0    0    0    0     0     0     8     8
##  [7,]    0    0    0    0    0    0    0    0    0     4     4     4     4
##  [8,]    0    0    0    0    0    0    0    0    7     7     7     7     7
##  [9,]    0    0    0    0    0    0    0    0    0     0     9     9     9
## [10,]    0    0    0    0    0    0    0    0   12    12    12    12    12
##       [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23]
##  [1,]    13    13    13    13    13    13    13    13    13     0
##  [2,]     6     6     6     0     2     2     0     3     3     3
##  [3,]    12    12    12    12    12    12    12    12    12     0
##  [4,]     7     7     0     7     7     7     7     7     7     7
##  [5,]     8     8     8     8     8     8     0     0     0     0
##  [6,]     8     8     8     8     8     8     0     0     0     0
##  [7,]     0     5     5     5     5     5     0     0     0     0
##  [8,]     7     7     0     1     0     2     2     0     0     0
##  [9,]     9     9     9     9     9     9     0     2     2     0
## [10,]    12    12    12    12    12    12    12     0     0     0
```

```r
set.seed(2)
k2 <- kmeans(mat_noncan_seed_cons_1, centers = 8)


# side annotations
col1 = colorRamp2(colors = c("white", "darkred"), breaks = c(0, 1))
col2 = colorRamp2( c("white", "darkblue"), breaks = c(0, 1))


binding_top_20 <- log10(noncan_seed[noncan_seed$no_8mer_pairing == T,]$scoreMax)
binding_top_20[binding_top_20 < quantile(binding_top_20, probs = seq(0, 1, 0.2))["80%"]] <- 0
binding_top_20[binding_top_20 >= quantile(binding_top_20, probs = seq(0, 1, 0.2))["80%"]] <- 1

free_energy_top_20 <- noncan_seed[noncan_seed$no_8mer_pairing == T,]$norm_free_energy
free_energy_top_20[free_energy_top_20  >= quantile(free_energy_top_20, probs = seq(0, 1, 0.2))["80%"]] <
free_energy_top_20[free_energy_top_20  < quantile(free_energy_top_20, probs = seq(0, 1, 0.2))["80%"]] <-
```

```
ra <- rowAnnotation(binding_strength = binding_top_20,
                    free_energy = free_energy_top_20,
                    col = list(binding_strength = col1,
                               free_energy = col2 ))


# plot heatmaps

set.seed(2)
Heatmap(mat_noncan_seed_cons_1, cluster_rows = F, cluster_columns = F, col = col_fun, split = k2$cluste
```




```
pdf(file = paste0(out,"Figure6G_Heatmap_without_seed.pdf"))
Heatmap(mat_noncan_seed_cons_1, cluster_rows = F, cluster_columns = F, col = col_fun, split = k2$cluste
dev.off()
```

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

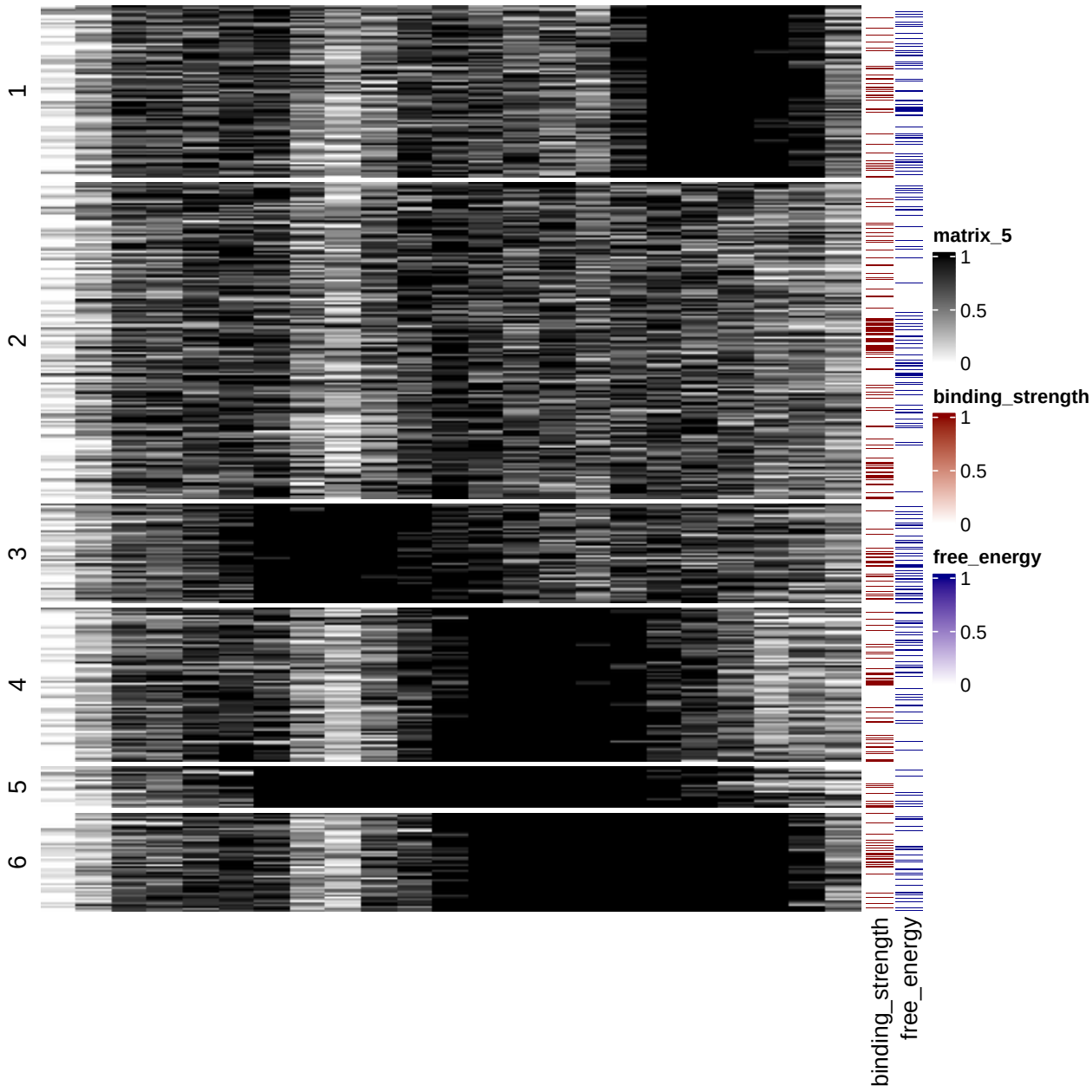
6.2 Heamap with parital binding in first 7 nt

```
# get mir181 pairing of canonical bound RNAs
mat_noncan_seed_2 <- noncan_seed %>%
  subset(!no_8mer_pairing) %>%
  dplyr::select(V2:V24) %>%
  as.matrix()

# color for heatmap
col_fun <- colorRamp2(colors = c("white", "black"), breaks = c(0, 1))

# make a matix with conseqtive 1 added up
mat_noncan_seed_cons_2 <- apply(mat_noncan_seed_2, 1, function(x){
  r = rle(x)
  r2 = r$lengths*r$values
  r3 = rep(r2,r$lengths)
  return(r3)
})

mat_noncan_seed_cons_2 <- t(mat_noncan_seed_cons_2)
mat_noncan_seed_cons_2[1:10,]
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
##  [1,]    0    0    5    5    5    5    5    0    7     7     7     7     7
##  [2,]    0    0    5    5    5    5    5    0    2     2     0     5     5
##  [3,]   11   11   11   11   11   11   11   11   11    11    11     0     0
##  [4,]    0    0    0    0    4    4    4    4    0     0     5     5     5
##  [5,]    0    3    3    3    0    3    3    3    0     0     0     0     6
##  [6,]    0    5    5    5    5    5    0    0    0     0     5     5     5
##  [7,]    0    2    2    0    1    0    6    6    6     6     6     6     0
##  [8,]    0    0    5    5    5    5    5    0    0     0     3     3     3
##  [9,]    3    3    3    0   14   14   14   14   14    14    14    14    14
## [10,]    0    0    6    6    6    6    6    6    0     0     0     0     0
##       [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23]
##  [1,]     7     7     0     4     4     4     4     0     2     2
##  [2,]     5     5     5     0     0     5     5     5     5     5
##  [3,]     7     7     7     7     7     7     7     0     2     2
##  [4,]     5     5     0     2     2     0     4     4     4     4
##  [5,]     6     6     6     6     6     0     4     4     4     4
##  [6,]     5     5     0     3     3     3     0     1     0     0
##  [7,]     2     2     0     6     6     6     6     6     6     0
##  [8,]     0     1     0     7     7     7     7     7     7     7
##  [9,]    14    14    14    14    14     0     4     4     4     4
## [10,]    10    10    10    10    10    10    10    10    10    10
```

```
set.seed(2)
k3 <- kmeans(mat_noncan_seed_cons_2, centers = 6)


# side annotations
col1 = colorRamp2(colors = c("white", "darkred"), breaks = c(0, 1))
```

```
col2 = colorRamp2( c("white", "darkblue"), breaks = c(0, 1))


binding_top_20 <- log10(noncan_seed[noncan_seed$no_8mer_pairing == F,]$scoreMax)
binding_top_20[binding_top_20 < quantile(binding_top_20, probs = seq(0, 1, 0.2))["80%"]] <- 0
binding_top_20[binding_top_20 >= quantile(binding_top_20, probs = seq(0, 1, 0.2))["80%"]] <- 1

free_energy_top_20 <- noncan_seed[noncan_seed$no_8mer_pairing == F,]$norm_free_energy
free_energy_top_20[free_energy_top_20  >= quantile(free_energy_top_20, probs = seq(0, 1, 0.2))["80%"]] <
free_energy_top_20[free_energy_top_20  < quantile(free_energy_top_20, probs = seq(0, 1, 0.2))["80%"]] <-

ra <- rowAnnotation(binding_strength = binding_top_20,
                    free_energy = free_energy_top_20,
                    col = list(binding_strength = col1,
                               free_energy = col2 ))


# plot heatmaps

set.seed(2)
Heatmap(mat_noncan_seed_cons_2, cluster_rows = F, cluster_columns = F, col = col_fun, split = k3$cluste
```



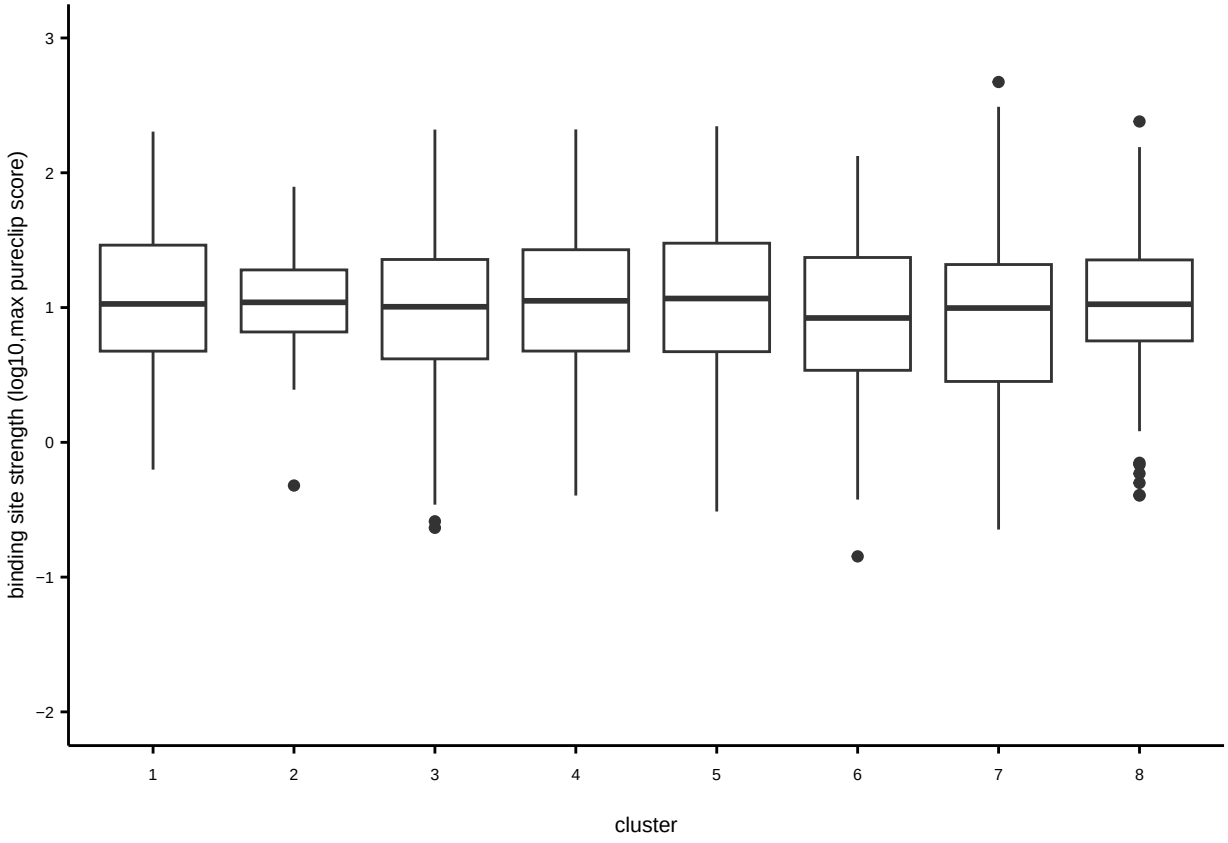

```
pdf(file = paste0(out,"Figure6E_Heatmap_without_seed.pdf"))
Heatmap(mat_noncan_seed_cons_2, cluster_rows = F, cluster_columns = F, col = col_fun, split = k3$cluste
dev.off()
```

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

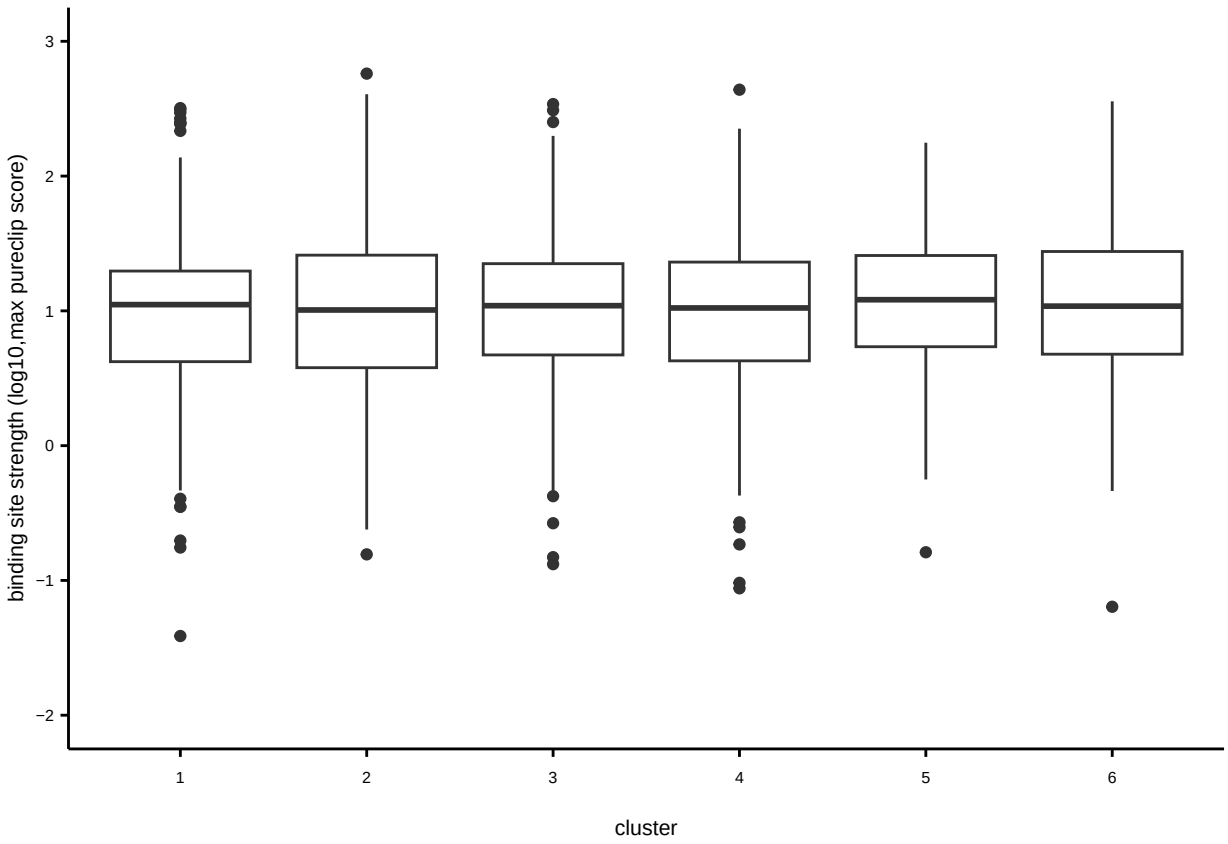
6.2.1 Bindingsite strength per cluster

```
noncan_seed$kmeans <- NA

noncan_seed[noncan_seed$no_8mer_pairing == T,]$kmeans <-
  k2$cluster
noncan_seed[noncan_seed$no_8mer_pairing == F,]$kmeans <-
```

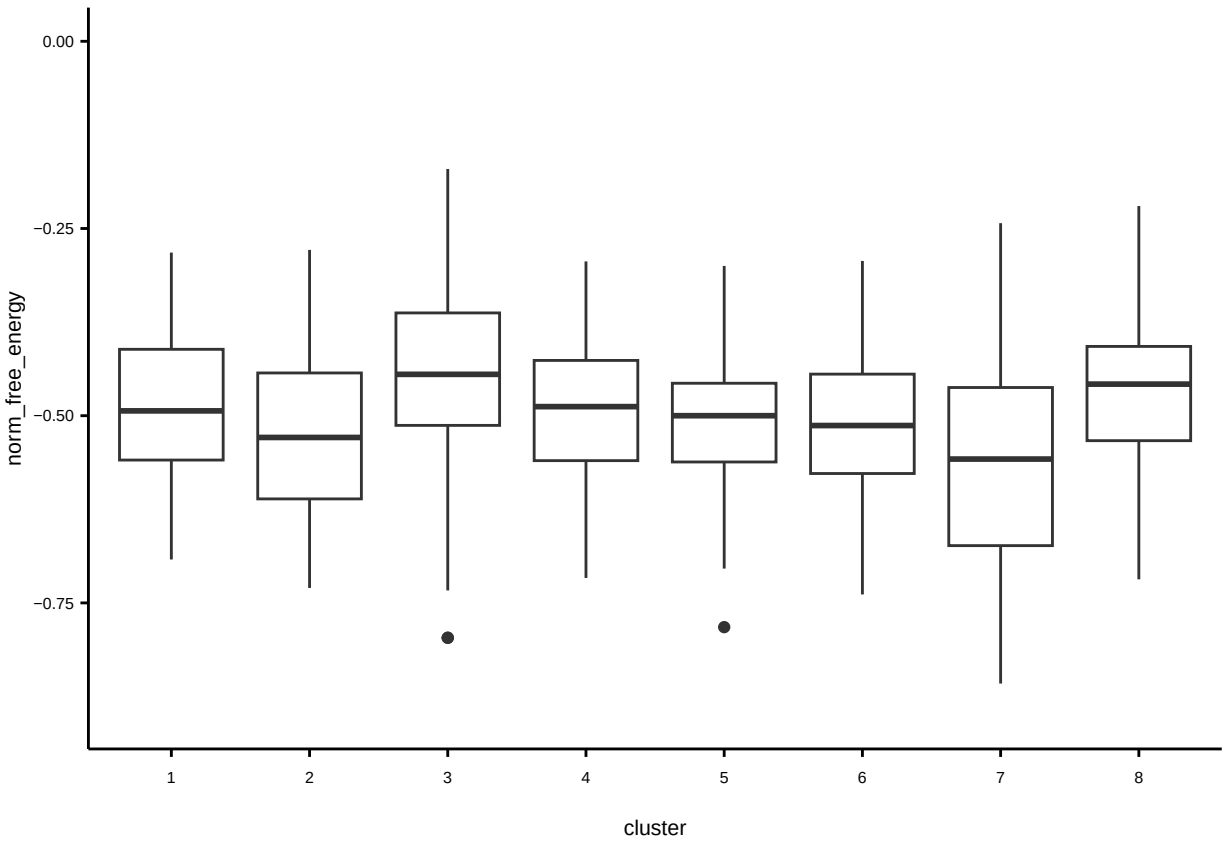
```
  (k3$cluster)

ggplot(noncan_seed[noncan_seed$no_8mer_pairing == T,], aes(as.character(kmeans), log10(scoreMax)))+
  geom_boxplot()+
  coord_cartesian(ylim= c(-2,3))+
  theme_paper()+
  xlab("cluster")+
  ylab("binding site strength (log10,max pureclip score)")
```



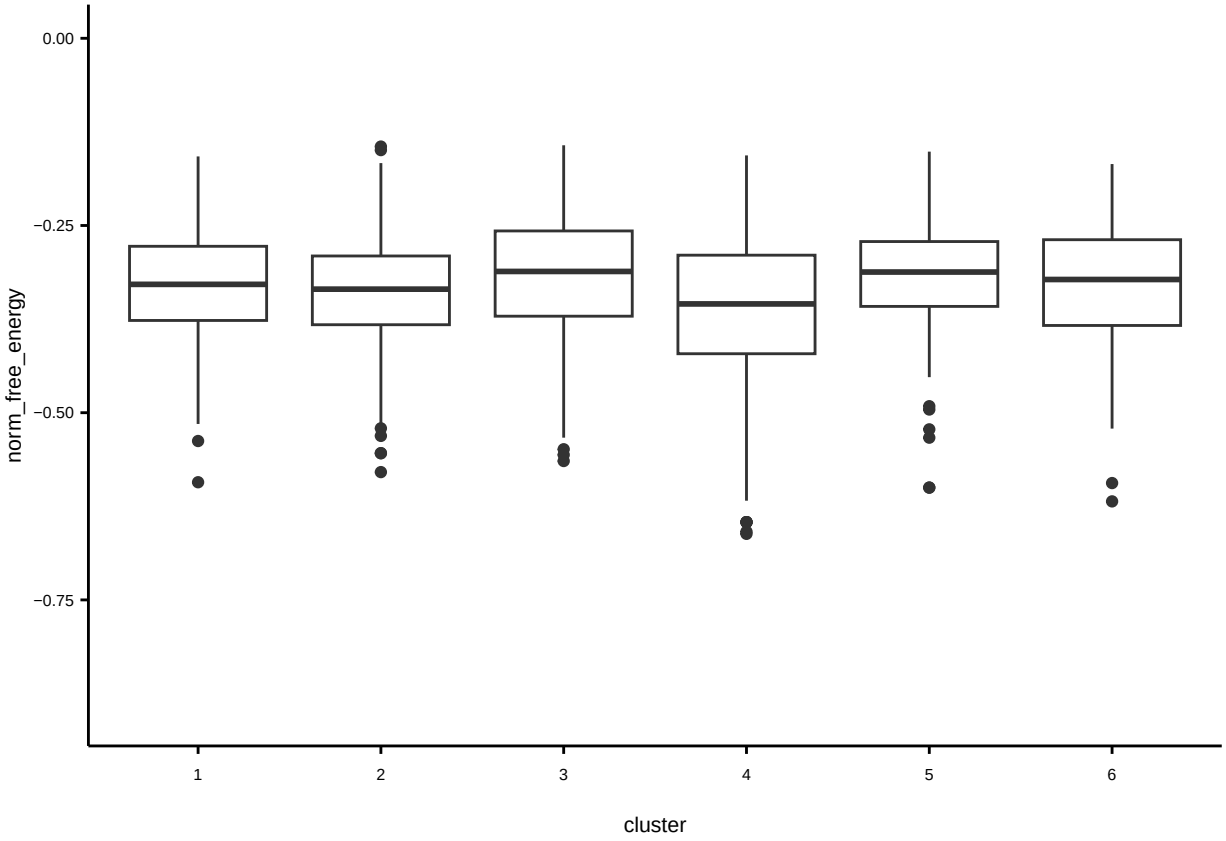

```
ggsave( filename = paste0(out, "SuppFigure6I_pureclip_score_per_cluster_noncan_noseed.pdf"), width = 6,


ggplot(noncan_seed[noncan_seed$no_8mer_pairing == F,], aes(as.character(kmeans), log10(scoreMax)))+
  geom_boxplot()+
  coord_cartesian(ylim= c(-2,3))+
  theme_paper()+
  xlab("cluster")+
  ylab("binding site strength (log10,max pureclip score)")
```




```
ggsave( filename = paste0(out, "SuppFigureS6J_pureclip_score_per_cluster_noncan_partseed.pdf"), width =

ggplot(noncan_seed[noncan_seed$no_8mer_pairing == T,], aes(as.character(kmeans), norm_free_energy))+
  geom_boxplot()+
  coord_cartesian(ylim= c(-0.9,0))+
  theme_paper()+
  xlab("cluster")
```




```
ggsave( filename = paste0(out, "SuppFigureS6F_min_free_energy_per_cluster_noncan_noseed.pdf"), width = 6

ggplot(noncan_seed[noncan_seed$no_8mer_pairing == F,], aes(as.character(kmeans), norm_free_energy))+
  geom_boxplot()+
  coord_cartesian(ylim= c(-0.9,0))+
  theme_paper()+
  xlab("cluster")
```

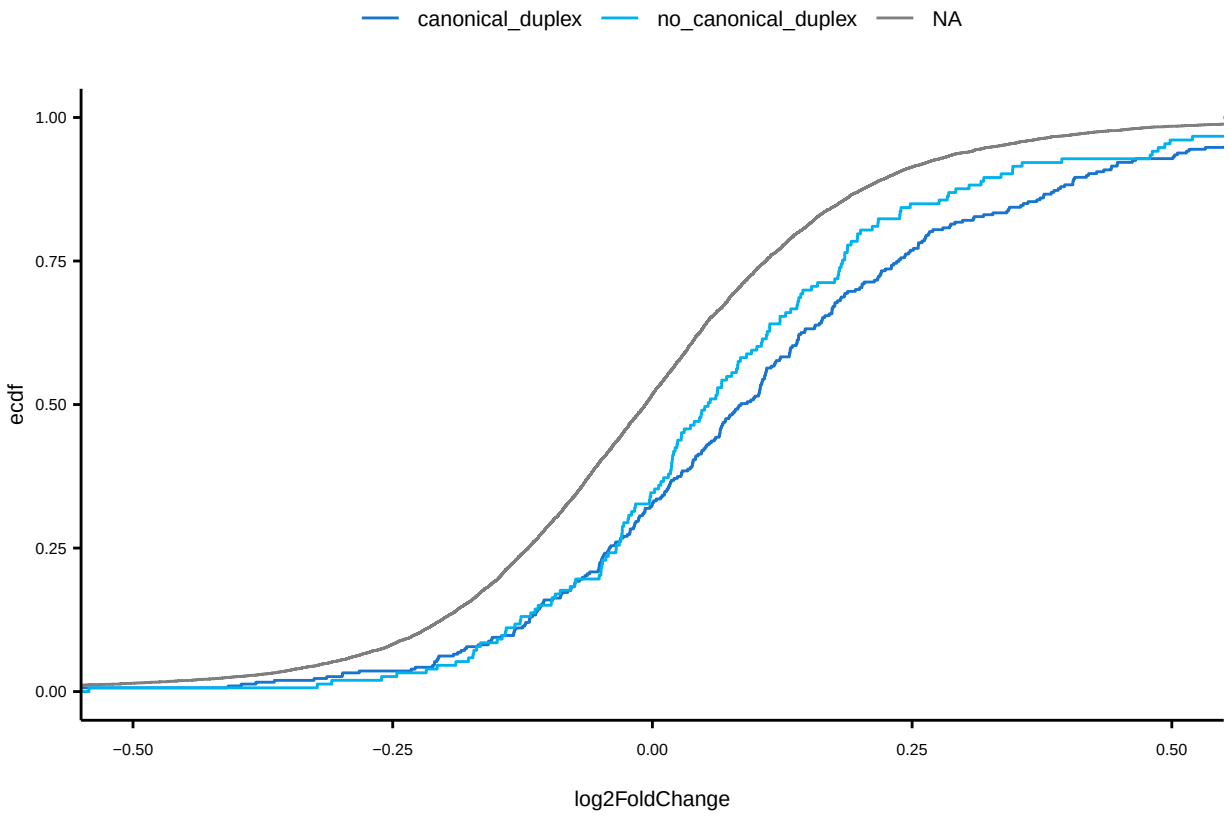



```
ggsave( filename = paste0(out, "SuppFigureS6G_min_free_energy_per_cluster_noncan_partseed.pdf"), width =
```

7 ECDFs

```
# -------------------------
# seed vs no seed
# -------------------------
rfp <- rfp %>%
  dplyr::select(c(log2FoldChange, Gene)) %>%
  mutate(seed_group = case_when(
    Gene %in% can_seed$geneID ~ "can_seed",
    Gene %in% noncan_seed$geneID ~ "non_can_seed",
    T ~ "no_target"
  ))



# -------------------------
# seed with vs seed without duplex
# -------------------------
rfp <- rfp %>% mutate(
  duplex_binds_seed = case_when(
    Gene %in% struct_bound_mir_df_6mer[which(struct_bound_mir_df_6mer$canonical_duplex == T),]$geneID ~
    Gene %in% struct_bound_mir_df_6mer[which(struct_bound_mir_df_6mer$canonical_duplex == F),]$geneID ~
  ))
```

```
ggplot(rfp %>% subset(seed_group != "non_can_seed"), aes(x = log2FoldChange, color = duplex_binds_seed )
  stat_ecdf( )+
  scale_color_manual( values = c("dodgerblue3", "deepskyblue2","darkgrey"))+
  coord_cartesian(xlim = c(-0.5, 0.5))+
  theme_paper()+
  theme(legend.position = "top")
```



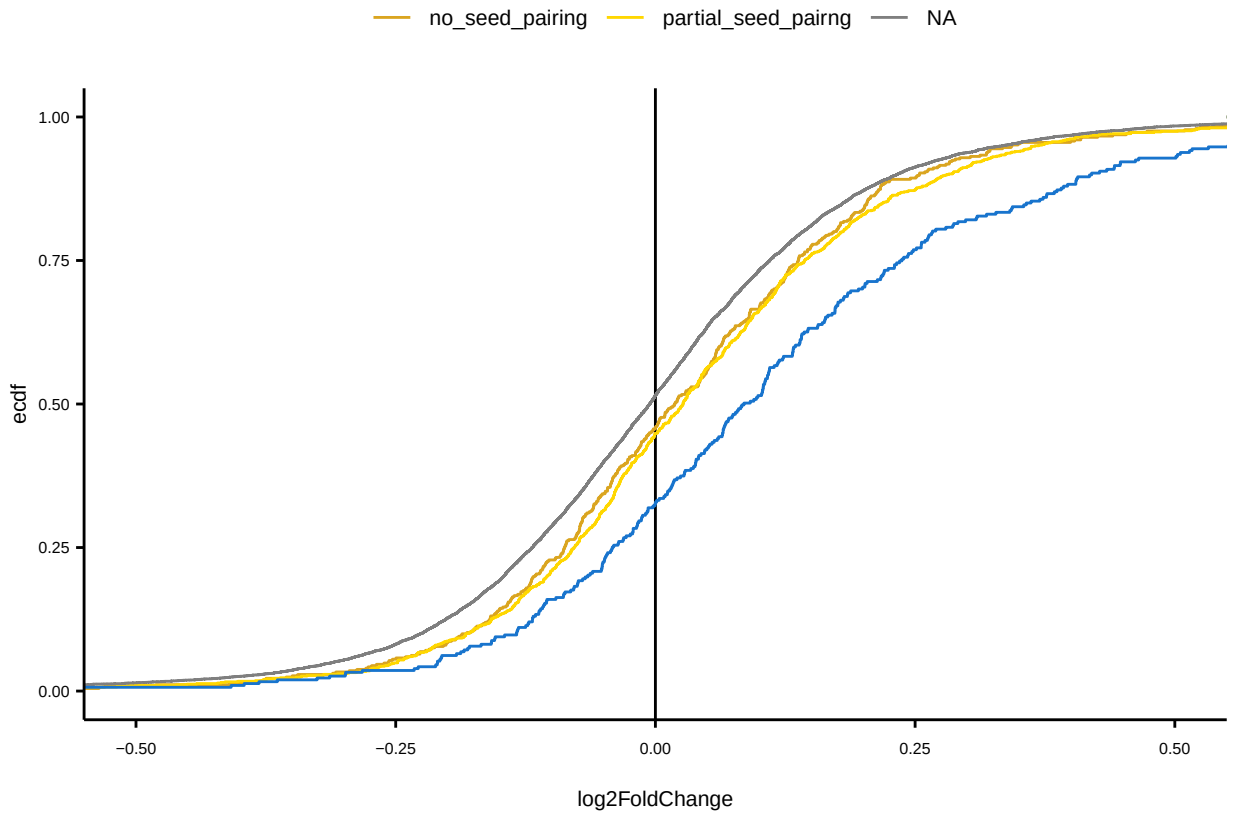

```
ggsave( filename = paste0(out, "SuppFigureS6d_ecdf_canonical_vs_non_canonical_duplex.pdf"), width = 5, h

ks.test(x = rfp %>% subset(seed_group == "no_target") %>% pull (log2FoldChange) %>% ecdf(.) %>% knots(.)
       y = rfp %>% subset(seed_group == "can_seed" & duplex_binds_seed == "canonical_duplex") %>% pull
```

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp %>% subset(seed_group == "no_target") %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) an
## D = 0.22244, p-value = 3.275e-13
## alternative hypothesis: two-sided
```

```
ks.test(x = rfp %>% subset(seed_group == "no_target") %>% pull (log2FoldChange) %>% ecdf(.) %>% knots(.)
       y = rfp %>% subset( duplex_binds_seed == "no_canonical_duplex") %>% pull (log2FoldChange) %>% ecd
```

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp %>% subset(seed_group == "no_target") %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) an
## D = 0.15094, p-value = 8.998e-06
## alternative hypothesis: two-sided
```

```
# ------------------------
# no seed no binding vs no seed partial binding
# ------------------------

rfp <- rfp %>% mutate(
  duplex_binds_seed = case_when(
    Gene %in% noncan_seed[which(noncan_seed$no_8mer_pairing == T),]$geneID ~ "no_seed_pairing",
    Gene %in% noncan_seed[which(noncan_seed$no_8mer_pairing == F),]$geneID ~ "partial_seed_pairng"
  ))


ggplot(rfp %>% subset(seed_group != "can_seed"), aes(x = log2FoldChange, color = duplex_binds_seed ))+
  geom_vline(xintercept = 0)+
  stat_ecdf( )+
  scale_color_manual( values = c("goldenrod", "gold",  "black"))+
  coord_cartesian(xlim = c(-0.5, 0.5))+
  stat_ecdf(data = rfp %>% subset(seed_group == "can_seed"), aes(x = log2FoldChange), color = "dodgerblu
  theme_paper()+
  theme(legend.position = "top")
```




```
ks.test(x = rfp %>% subset(seed_group == "no_target") %>% pull (log2FoldChange) %>% ecdf(.) %>% knots(.
        y = rfp %>% subset(seed_group != "can_seed" & duplex_binds_seed == "partial_seed_pairng") %>% pul
```

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp %>% subset(seed_group == "no_target") %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) a
## D = 0.088671, p-value = 1.338e-06
```

```
## alternative hypothesis: two-sided
```

```
ks.test(x = rfp %>% subset(seed_group == "no_target") %>% pull (log2FoldChange) %>% ecdf(.) %>% knots(.)
       y = rfp %>% subset(seed_group != "can_seed" & duplex_binds_seed == "no_seed_pairing") %>% pull (l
```

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp %>% subset(seed_group == "no_target") %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) ar
## D = 0.083466, p-value = 0.004946
## alternative hypothesis: two-sided
```

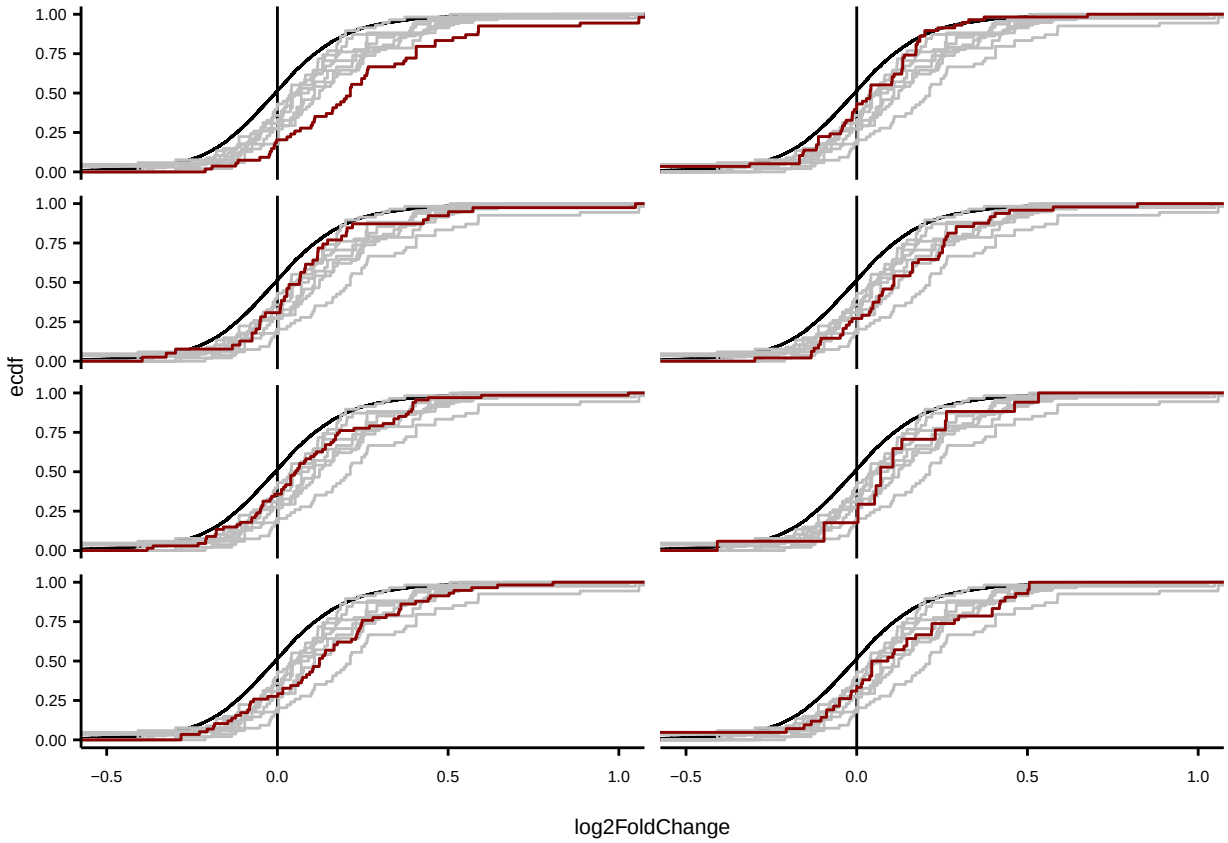
```
ks.test(x = rfp %>% subset(seed_group == "no_target") %>% pull (log2FoldChange) %>% ecdf(.) %>% knots(.)
       y = rfp %>% subset(seed_group == "can_seed") %>% pull (log2FoldChange) %>% ecdf(.) %>% knots(.))
```

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp %>% subset(seed_group == "no_target") %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) ar
## D = 0.22244, p-value = 3.275e-13
## alternative hypothesis: two-sided
```

```
ggsave( filename = paste0(out, "Figure6B_ecdf_partil_vs_no_seed_binding.pdf"), width = 5, height = 6, u
```

```
# ------------------------
# clusters seed + duplex
# ------------------------
rfp_non_targets <- rfp %>% subset(seed_group == "no_target")

rfp_can <- left_join(rfp,
                     can_seed %>% select("geneID", "kmeans"),
                     by = c(Gene = "geneID"))  %>%
  subset(seed_group == "can_seed")

rfp_can_bg_gg_1 <- rfp_can %>% subset(kmeans == 1)
rfp_can_bg_gg_2 <- rfp_can %>% subset(kmeans == 2)
rfp_can_bg_gg_3 <- rfp_can %>% subset(kmeans == 3)
rfp_can_bg_gg_4 <- rfp_can %>% subset(kmeans == 4)
rfp_can_bg_gg_5 <- rfp_can %>% subset(kmeans == 5)
rfp_can_bg_gg_6 <- rfp_can %>% subset(kmeans == 6)
rfp_can_bg_gg_7 <- rfp_can %>% subset(kmeans == 7)
rfp_can_bg_gg_8 <- rfp_can %>% subset(kmeans == 8)

rfp_can_bg_gg_1$kmeans <- NULL
rfp_can_bg_gg_2$kmeans <- NULL
rfp_can_bg_gg_3$kmeans <- NULL
rfp_can_bg_gg_4$kmeans <- NULL
rfp_can_bg_gg_5$kmeans <- NULL
rfp_can_bg_gg_6$kmeans <- NULL
rfp_can_bg_gg_7$kmeans <- NULL
rfp_can_bg_gg_8$kmeans <- NULL
rfp_non_targets$kmeans <- NULL



ggplot(rfp_can, aes(x = log2FoldChange))+
```

```
  geom_vline(xintercept = 0)+
  stat_ecdf(data = rfp_non_targets , aes(x = log2FoldChange), color = "black")+
  stat_ecdf(data = rfp_can_bg_gg_1, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_can_bg_gg_2, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_can_bg_gg_3, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_can_bg_gg_4, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_can_bg_gg_5, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_can_bg_gg_6, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_can_bg_gg_7, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_can_bg_gg_8, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf( color = "darkred" )+
  facet_wrap(~kmeans, ncol = 2)+
  coord_cartesian(xlim = c(-0.5, 1))+
  theme_paper()+
  theme(legend.position = "top")+
  theme(
  strip.text.x = element_blank()
)
```

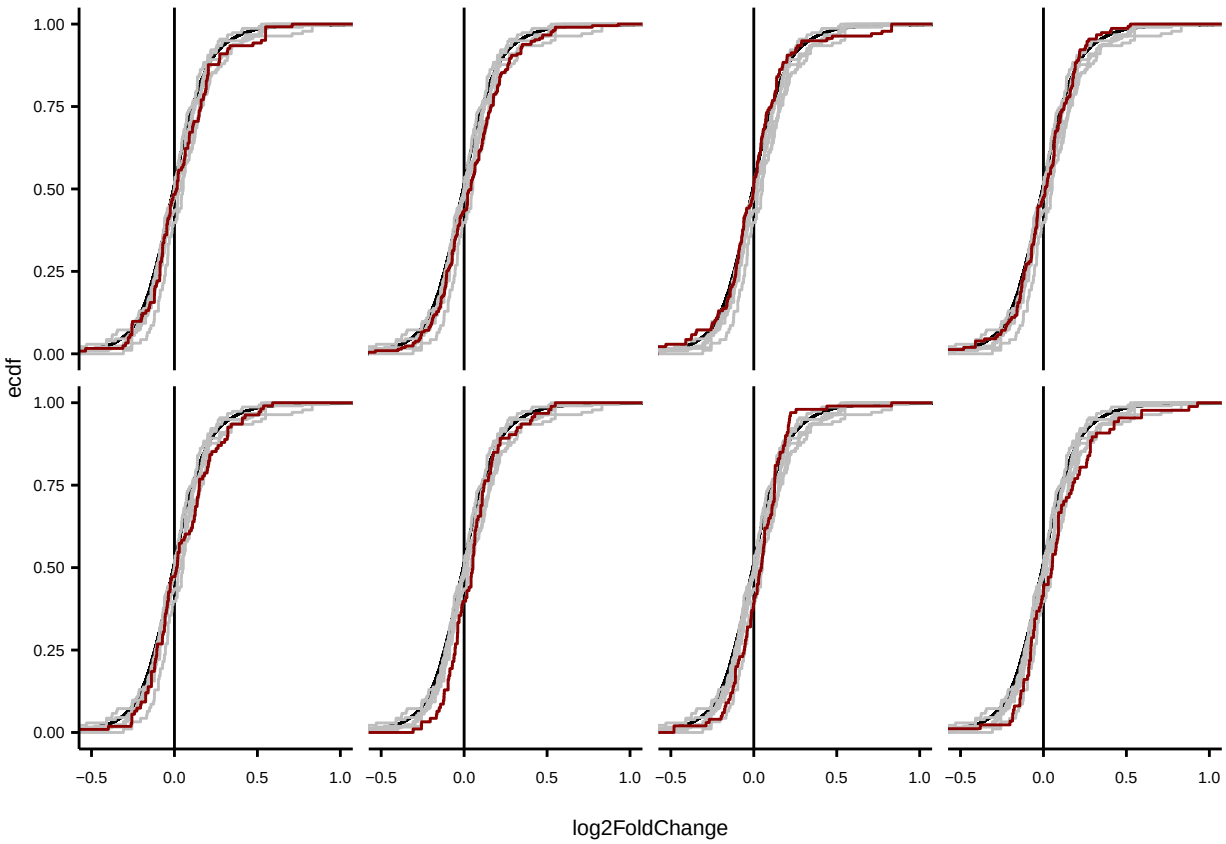



```
ggsave( filename = paste0(out, "Figure6d_ecdf_clusters_can.pdf"), width = 5, height = 8, units = "cm")

for(x in 1:8){
  print(x)
  print(ks.test(x = rfp_non_targets %>% pull (log2FoldChange) %>% ecdf(.) %>% knots(.) ,
       y = rfp_can %>% subset(kmeans == x) %>% pull (log2FoldChange) %>% ecdf(.) %>% knots(.)))
}
```

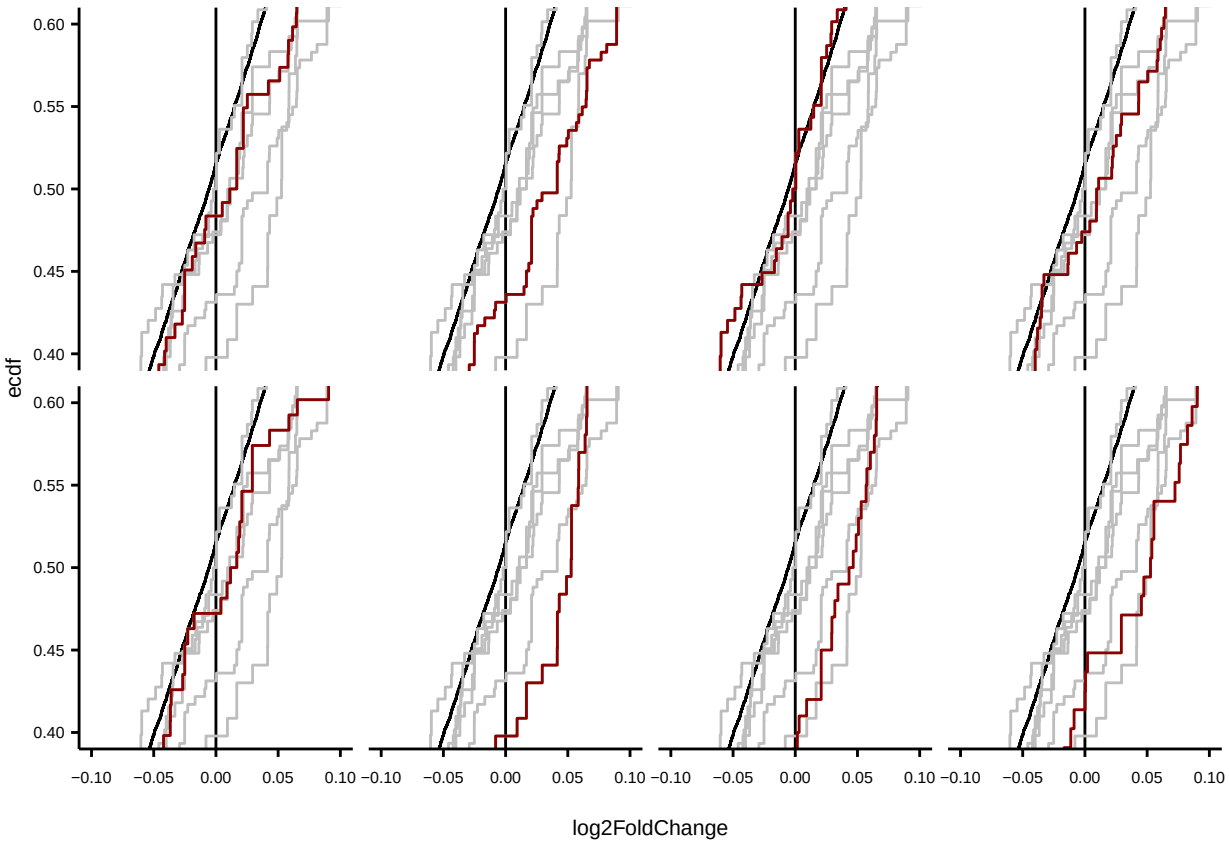
```
## [1] 1
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp_non_targets %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) and rfp_can %>% subset(kmean
## D = 0.39742, p-value = 1.433e-06
## alternative hypothesis: two-sided
##
## [1] 2
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp_non_targets %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) and rfp_can %>% subset(kmean
## D = 0.18509, p-value = 0.1219
## alternative hypothesis: two-sided
##
## [1] 3
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp_non_targets %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) and rfp_can %>% subset(kmean
## D = 0.24392, p-value = 0.02213
## alternative hypothesis: two-sided
##
## [1] 4
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp_non_targets %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) and rfp_can %>% subset(kmean
## D = 0.29254, p-value = 0.001558
## alternative hypothesis: two-sided
##
## [1] 5
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp_non_targets %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) and rfp_can %>% subset(kmean
## D = 0.17696, p-value = 0.05407
## alternative hypothesis: two-sided
##
## [1] 6
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp_non_targets %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) and rfp_can %>% subset(kmean
## D = 0.40958, p-value = 0.02567
## alternative hypothesis: two-sided
##
## [1] 7
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp_non_targets %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) and rfp_can %>% subset(kmean
## D = 0.33007, p-value = 2.552e-05
```

```
## alternative hypothesis: two-sided
##
## [1] 8
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp_non_targets %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) and rfp_can %>% subset(kmean
## D = 0.23124, p-value = 0.05938
## alternative hypothesis: two-sided
```

```
# ------------------------
# clusters no seed
# ------------------------
rfp_nocan_noseed <- left_join(rfp,
                              noncan_seed, by = c(Gene = "geneID"),
                              suffix = c(".can", ".noncan")) %>%
  subset((seed_group == "non_can_seed") & duplex_binds_seed == "no_seed_pairing" )

rfp_nocan_noseed_bg_gg <- rfp_nocan_noseed
rfp_nocan_noseed_bg_gg_1 <- rfp_nocan_noseed_bg_gg %>% subset(kmeans == 1)
rfp_nocan_noseed_bg_gg_2 <- rfp_nocan_noseed_bg_gg %>% subset(kmeans == 2)
rfp_nocan_noseed_bg_gg_3 <- rfp_nocan_noseed_bg_gg %>% subset(kmeans == 3)
rfp_nocan_noseed_bg_gg_4 <- rfp_nocan_noseed_bg_gg %>% subset(kmeans == 4)
rfp_nocan_noseed_bg_gg_5 <- rfp_nocan_noseed_bg_gg %>% subset(kmeans == 5)
rfp_nocan_noseed_bg_gg_6 <- rfp_nocan_noseed_bg_gg %>% subset(kmeans == 6)


rfp_nocan_noseed_bg_gg_1$kmeans <- NULL
rfp_nocan_noseed_bg_gg_2$kmeans <- NULL
rfp_nocan_noseed_bg_gg_3$kmeans <- NULL
rfp_nocan_noseed_bg_gg_4$kmeans <- NULL
rfp_nocan_noseed_bg_gg_5$kmeans <- NULL
rfp_nocan_noseed_bg_gg_6$kmeans <- NULL


ggplot(rfp_nocan_noseed, aes(x = log2FoldChange, colour = as.character(rfp_nocan_noseed$kmeans.noncan))
geom_vline(xintercept = 0)+
  stat_ecdf(data = rfp_non_targets , aes(x = log2FoldChange), color = "black")+
  stat_ecdf(data = rfp_nocan_noseed_bg_gg_1, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_nocan_noseed_bg_gg_2, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_nocan_noseed_bg_gg_3, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_nocan_noseed_bg_gg_4, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_nocan_noseed_bg_gg_5, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_nocan_noseed_bg_gg_6, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf( color = "darkred" )+
  facet_wrap(~kmeans, ncol = 4)+
  coord_cartesian(xlim = c(-0.5, 1))+
  theme_paper()+
  theme(legend.position = "top")+
  theme(
  strip.text.x = element_blank()
)
```




```
ggsave( filename = paste0(out, "Figure6H_ecdf_clusters_nocan_noseed.pdf"), width = 9, height = 5, units

# zoom-ins
ggplot(rfp_nocan_noseed, aes(x = log2FoldChange, colour = as.character(rfp_nocan_noseed$kmeans.noncan))
geom_vline(xintercept = 0)+
  stat_ecdf(data = rfp_non_targets , aes(x = log2FoldChange), color = "black")+
  stat_ecdf(data = rfp_nocan_noseed_bg_gg_1, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_nocan_noseed_bg_gg_2, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_nocan_noseed_bg_gg_3, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_nocan_noseed_bg_gg_4, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_nocan_noseed_bg_gg_5, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_nocan_noseed_bg_gg_6, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf( color = "darkred" )+
  facet_wrap(~kmeans, ncol = 4)+
  coord_cartesian(xlim = c(-0.1, 0.1), ylim = c(0.4, 0.6))+
  theme_paper()+
  theme(legend.position = "top")+
  theme(
  strip.text.x = element_blank()
)
```




```
ggsave( filename = paste0(out, "Figure6H_ecdf_clusters_nocan_noseed_zoomIn.pdf"), width = 9, height = 5
```

```
# komogronov test
for(i in 1:8){
  print(i)
  print(ks.test(x = rfp_non_targets %>% pull (log2FoldChange) %>% ecdf(.) %>% knots(.) ,
       y = rfp_nocan_noseed %>% subset(kmeans == i) %>% pull (log2FoldChange) %>% ecdf(.) %>% knots(.)))
}
```

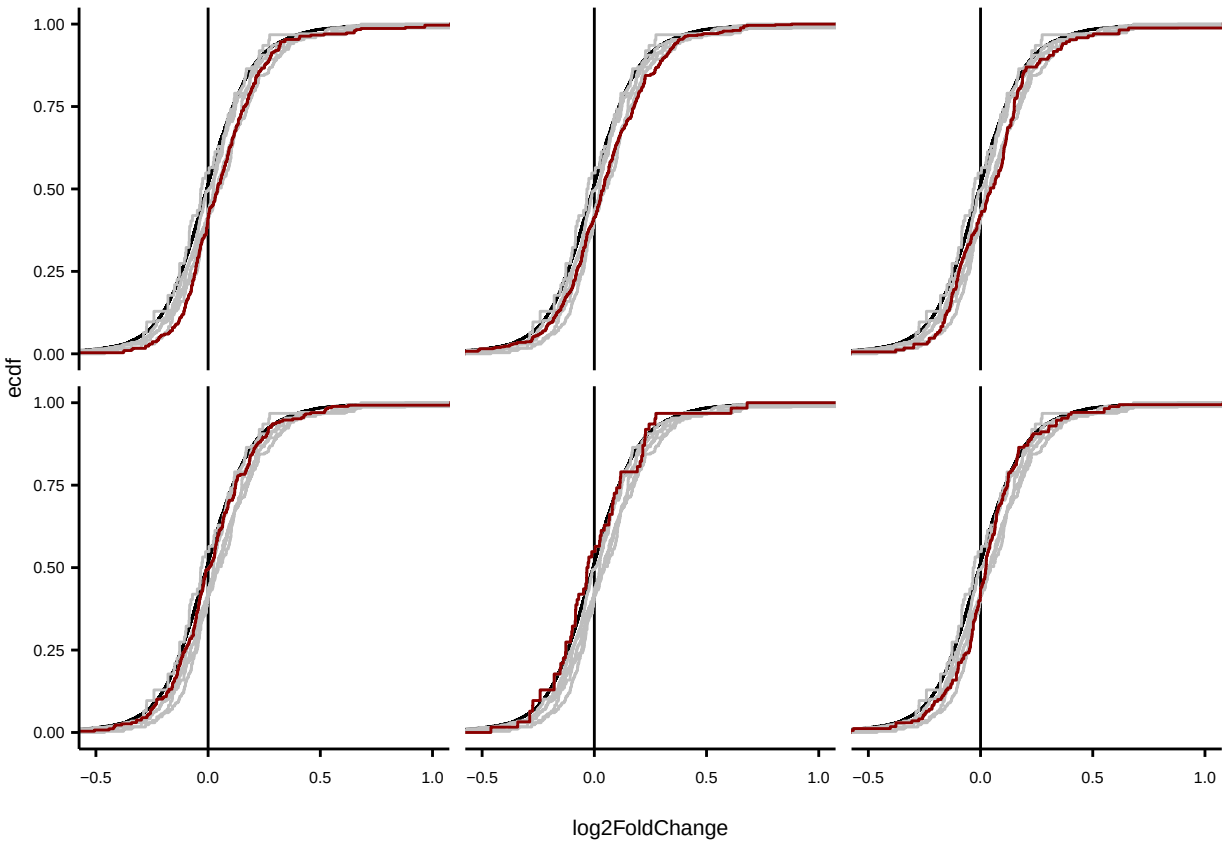
```
## [1] 1
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp_non_targets %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) and rfp_nocan_noseed %>% sub
## D = 0.085901, p-value = 0.5406
## alternative hypothesis: two-sided
##
## [1] 2
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp_non_targets %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) and rfp_nocan_noseed %>% sub
## D = 0.12741, p-value = 0.01654
## alternative hypothesis: two-sided
##
```

```
## [1] 3
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp_non_targets %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) and rfp_nocan_noseed %>% sub
## D = 0.039337, p-value = 0.9955
## alternative hypothesis: two-sided
##
## [1] 4
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp_non_targets %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) and rfp_nocan_noseed %>% sub
## D = 0.086528, p-value = 0.3278
## alternative hypothesis: two-sided
##
## [1] 5
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp_non_targets %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) and rfp_nocan_noseed %>% sub
## D = 0.17478, p-value = 0.009144
## alternative hypothesis: two-sided
##
## [1] 6
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp_non_targets %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) and rfp_nocan_noseed %>% sub
## D = 0.1866, p-value = 0.009791
## alternative hypothesis: two-sided
##
## [1] 7
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp_non_targets %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) and rfp_nocan_noseed %>% sub
## D = 0.13956, p-value = 0.06697
## alternative hypothesis: two-sided
##
## [1] 8
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp_non_targets %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) and rfp_nocan_noseed %>% sub
## D = 0.14969, p-value = 0.07445
## alternative hypothesis: two-sided
```

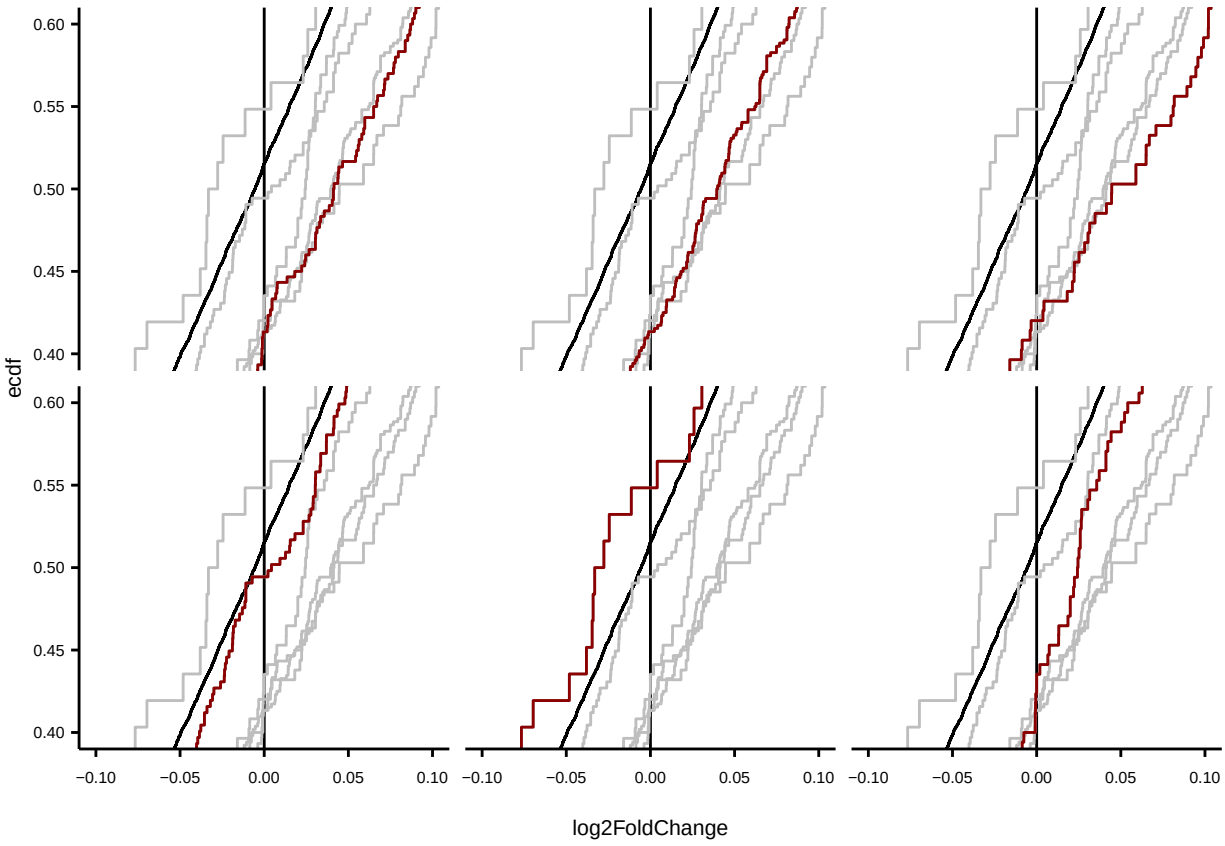
```
# n per cluster

table(rfp_nocan_noseed$kmeans)
```

```
##
##   1   2   3   4   5   6   7   8
## 122 211 138 154 108  93 100  87
```

```r
# ------------------------
# clusters partial seed
# ------------------------

rfp_nocan_partseed <- left_join(rfp, noncan_seed, by = c(Gene = "geneID"), suffix = c(".can", ".noncan")
  subset((seed_group == "non_can_seed") & duplex_binds_seed == "partial_seed_pairng" )

rfp_nocan_partseed_bg_gg_1 <- rfp_nocan_partseed %>% subset(kmeans == 1)
rfp_nocan_partseed_bg_gg_2 <- rfp_nocan_partseed %>% subset(kmeans == 2)
rfp_nocan_partseed_bg_gg_3 <- rfp_nocan_partseed %>% subset(kmeans == 3)
rfp_nocan_partseed_bg_gg_4 <- rfp_nocan_partseed %>% subset(kmeans == 4)
rfp_nocan_partseed_bg_gg_5 <- rfp_nocan_partseed %>% subset(kmeans == 5)
rfp_nocan_partseed_bg_gg_6 <- rfp_nocan_partseed %>% subset(kmeans == 6)
rfp_nocan_partseed_bg_gg_7 <- rfp_nocan_partseed %>% subset(kmeans == 7)
rfp_nocan_partseed_bg_gg_8 <- rfp_nocan_partseed %>% subset(kmeans == 8)


rfp_nocan_partseed_bg_gg_1$kmeans <- NULL
rfp_nocan_partseed_bg_gg_2$kmeans <- NULL
rfp_nocan_partseed_bg_gg_3$kmeans <- NULL
rfp_nocan_partseed_bg_gg_4$kmeans <- NULL
rfp_nocan_partseed_bg_gg_5$kmeans <- NULL
rfp_nocan_partseed_bg_gg_6$kmeans <- NULL
rfp_nocan_partseed_bg_gg_7$kmeans <- NULL
rfp_nocan_partseed_bg_gg_8$kmeans <- NULL


ggplot(rfp_nocan_partseed, aes(x = log2FoldChange))+
geom_vline(xintercept = 0)+
  stat_ecdf(data = rfp_non_targets , aes(x = log2FoldChange), color = "black")+
  stat_ecdf(data = rfp_nocan_partseed_bg_gg_1, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_nocan_partseed_bg_gg_2, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_nocan_partseed_bg_gg_3, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_nocan_partseed_bg_gg_4, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_nocan_partseed_bg_gg_5, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_nocan_partseed_bg_gg_6, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf( color = "darkred" )+
  facet_wrap(~kmeans, ncol = 3)+
  coord_cartesian(xlim = c(-0.5, 1))+
  theme_paper()+
  theme(legend.position = "top")+
  theme(
  strip.text.x = element_blank()
)
```




```
ggsave( filename = paste0(out, "Figure6F_ecdf_clusters_nocan_partseed.pdf"), width = 7, height = 5, uni

# zoom ins
ggplot(rfp_nocan_partseed, aes(x = log2FoldChange))+
geom_vline(xintercept = 0)+
  stat_ecdf(data = rfp_non_targets , aes(x = log2FoldChange), color = "black")+
  stat_ecdf(data = rfp_nocan_partseed_bg_gg_1, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_nocan_partseed_bg_gg_2, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_nocan_partseed_bg_gg_3, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_nocan_partseed_bg_gg_4, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_nocan_partseed_bg_gg_5, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf(data = rfp_nocan_partseed_bg_gg_6, aes(x = log2FoldChange), color = "grey")+
  stat_ecdf( color = "darkred" )+
  facet_wrap(~kmeans, ncol = 3)+
  coord_cartesian(xlim = c(-0.1, 0.1), ylim = c(0.4, 0.6))+
  theme_paper()+
  theme(legend.position = "top")+
  theme(
  strip.text.x = element_blank())
```




```
ggsave( filename = paste0(out, "Figure6F_ecdf_clusters_nocan_partseed_zoomIns.pdf"), width = 7, height =

# kmogornov test

for(i in 1:6){
  print(i)
  print(ks.test(x = rfp_non_targets %>% pull (log2FoldChange) %>% ecdf(.) %>% knots(.) ,
       y = rfp_nocan_partseed %>% subset(kmeans == i) %>% pull (log2FoldChange) %>% ecdf(.) %>% knots(.)
}
```

```
## [1] 1
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp_non_targets %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) and rfp_nocan_partseed %>% s
## D = 0.13706, p-value = 0.000177
## alternative hypothesis: two-sided
##
## [1] 2
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp_non_targets %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) and rfp_nocan_partseed %>% s
## D = 0.10473, p-value = 0.0002815
## alternative hypothesis: two-sided
##
## [1] 3
```

```
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp_non_targets %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) and rfp_nocan_partseed %>% s
## D = 0.15518, p-value = 0.001483
## alternative hypothesis: two-sided
##
## [1] 4
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp_non_targets %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) and rfp_nocan_partseed %>% s
## D = 0.052804, p-value = 0.5693
## alternative hypothesis: two-sided
##
## [1] 5
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp_non_targets %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) and rfp_nocan_partseed %>% s
## D = 0.07831, p-value = 0.8843
## alternative hypothesis: two-sided
##
## [1] 6
##
##  Asymptotic two-sample Kolmogorov-Smirnov test
##
## data:  rfp_non_targets %>% pull(log2FoldChange) %>% ecdf(.) %>% knots(.) and rfp_nocan_partseed %>% s
## D = 0.14867, p-value = 0.003321
## alternative hypothesis: two-sided
```

```
# numbers

table(rfp_nocan_partseed$kmeans)
```

```
##
##   1   2   3   4   5   6
## 300 520 169 267  62 170
```

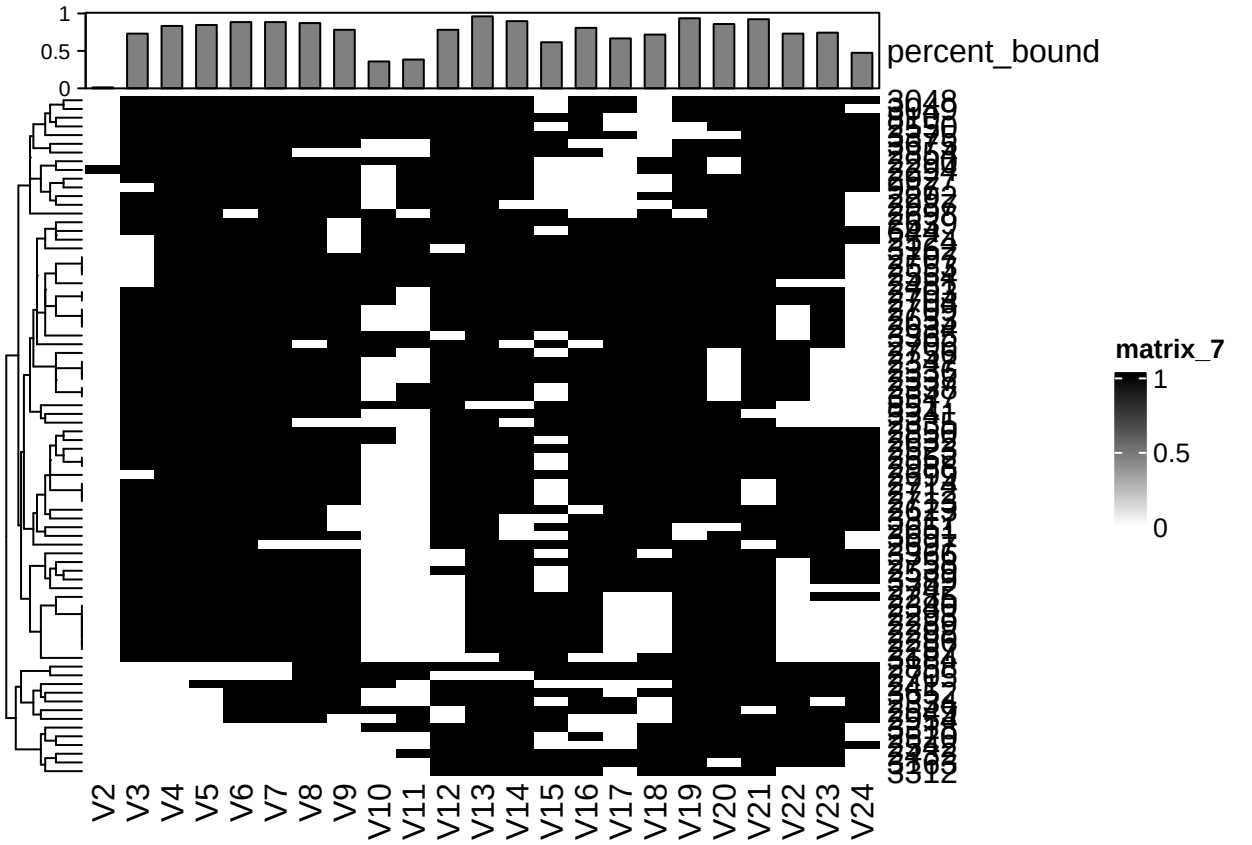
8 Check MMSAT4 / MurSatRep1 3'contribution

```
bs_on_rep <- readRDS(paste0(here, "/Figure4/04_MMsat4/bs_with_rep_transcript.rds"))

mmsat4 <- bs_on_rep[bs_on_rep$repName == "MMSAT4"]
mursatrep1 <- bs_on_rep[bs_on_rep$repName == "MurSatRep1"]

mmsat4_st <- struct_bound_mir_df %>% subset(mir181BS_ID %in% mmsat4$mir181BS_ID)
mursatrep1_st <- struct_bound_mir_df %>% subset(mir181BS_ID %in% mursatrep1$mir181BS_ID)
```

8.1 MMSAT4

```
mat_mmsat4 <- mmsat4_st %>% select(V2:V24) %>%
  as.matrix()
```

```
contr_mmsat4 <- colSums(mat_mmsat4) / nrow(mat_mmsat4)

ca <- HeatmapAnnotation(percent_bound = anno_barplot(contr_mmsat4))

Heatmap(mat_mmsat4, cluster_rows = T, cluster_columns = F, col = col_fun, top_annotation = ca )
```

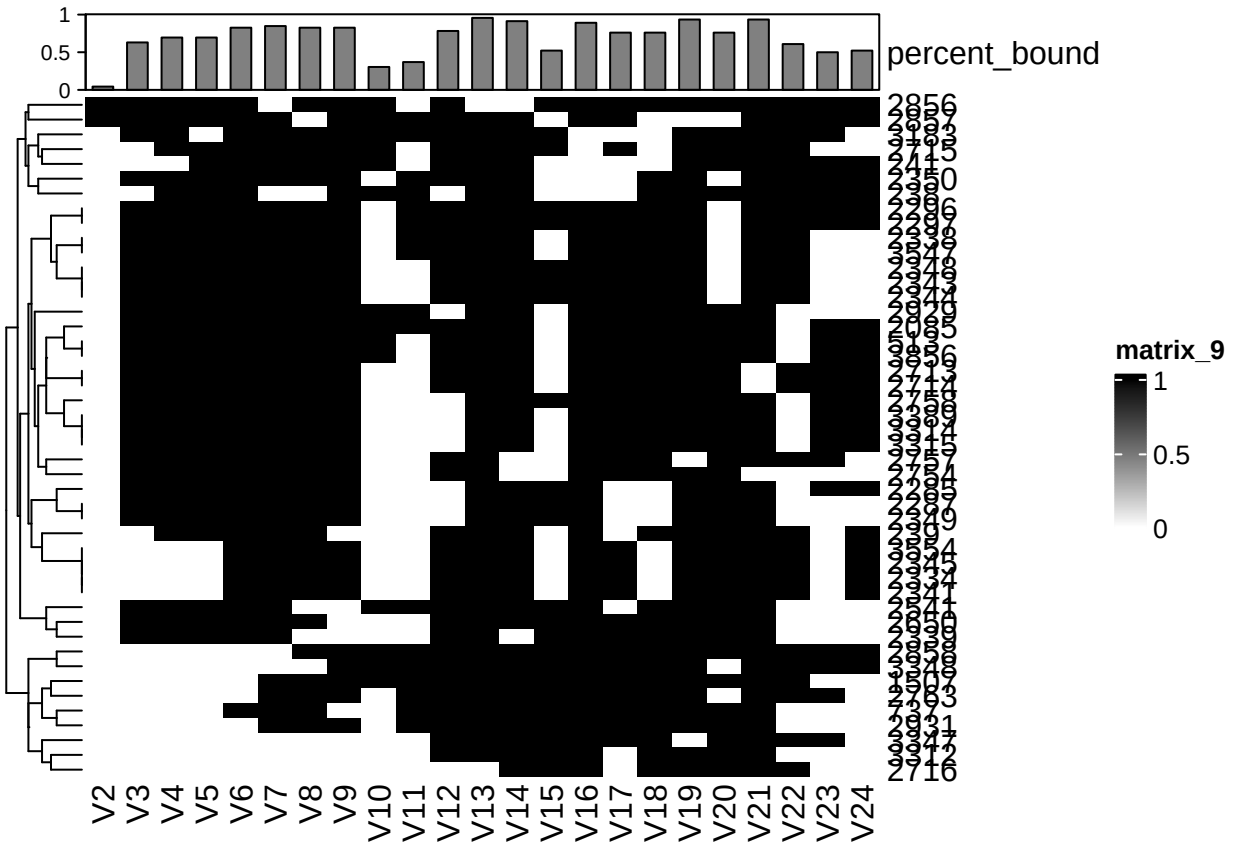



```
pdf(file = paste0(out,"SuppFigureS6K_Heatmap_mmsat4.pdf"))
Heatmap(mat_mmsat4, cluster_rows = T, cluster_columns = F, col = col_fun, top_annotation = ca )
dev.off()
```

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

8.2 MurSatRep1

```
mat_mursatrep1 <- mursatrep1_st %>% select(V2:V24) %>%
  as.matrix()
contr_mursatrep1 <- colSums(mat_mursatrep1) / nrow(mat_mursatrep1)

ca <- HeatmapAnnotation(percent_bound = anno_barplot(contr_mursatrep1))

Heatmap(mat_mursatrep1, cluster_rows = T, cluster_columns = F, col = col_fun, top_annotation = ca)
```




```
pdf(file = paste0(out,"SuppFigureS6L_Heatmap_mursatrep1.pdf"))
Heatmap(mat_mursatrep1, cluster_rows = T, cluster_columns = F, col = col_fun, top_annotation = ca)
dev.off()
```

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