

RNAhybrid_heatmaps_fig2

Nikita Verheyden

2023-04-12

Setup

directory

```
setwd("D:/Krueger_Lab/Publications/miR181_paper/Figure2/RNAhybrid")  
  
set.seed(123)
```

packages

```
source("D:/Krueger_Lab/Publications/miR181_paper_v21022023/Figure_theme/theme_paper.R")  
library(BSgenome.Mmusculus.UCSC.mm10)
```

```
## Loading required package: BSgenome  
## Loading required package: BiocGenerics  
##  
## Attaching package: 'BiocGenerics'  
## The following objects are masked from 'package:stats':  
##  
##   IQR, mad, sd, var, xtabs  
## The following objects are masked from 'package:base':  
##  
##   anyDuplicated, aperm, append, as.data.frame, basename, cbind,  
##   colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,  
##   get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,  
##   match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,  
##   Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,  
##   table, tapply, union, unique, unsplit, which.max, which.min  
## Loading required package: S4Vectors  
## Loading required package: stats4  
##  
## Attaching package: 'S4Vectors'  
## The following objects are masked from 'package:base':  
##  
##   expand.grid, I, unname  
## Loading required package: IRanges  
##
```

```

## Attaching package: 'IRanges'
## The following object is masked from 'package:grDevices':
##
##     windows
## Loading required package: GenomeInfoDb
## Loading required package: GenomicRanges
## Loading required package: Biostrings
## Loading required package: XVector
##
## Attaching package: 'Biostrings'
## The following object is masked from 'package:base':
##
##     strsplit
## Loading required package: rtracklayer
library(dplyr)

##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:Biostrings':
##
##     collapse, intersect, setdiff, setequal, union
## The following object is masked from 'package:XVector':
##
##     slice
## The following objects are masked from 'package:GenomicRanges':
##
##     intersect, setdiff, union
## The following object is masked from 'package:GenomeInfoDb':
##
##     intersect
## The following objects are masked from 'package:IRanges':
##
##     collapse, desc, intersect, setdiff, slice, union
## The following objects are masked from 'package:S4Vectors':
##
##     first, intersect, rename, setdiff, setequal, union
## The following objects are masked from 'package:BiocGenerics':
##
##     combine, intersect, setdiff, union
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

```

```
library(ggplot2)
library(seqinr)
```

```
##
## Attaching package: 'seqinr'

## The following object is masked from 'package:dplyr':
##
##     count

## The following object is masked from 'package:Biostrings':
##
##     translate
```

```
library(circlize)
```

```
## =====
## circlize version 0.4.15
## CRAN page: https://cran.r-project.org/package=circlize
## Github page: https://github.com/jokergoo/circlize
## Documentation: https://jokergoo.github.io/circlize\_book/book/
##
## If you use it in published research, please cite:
## Gu, Z. circlize implements and enhances circular visualization
##   in R. Bioinformatics 2014.
##
## This message can be suppressed by:
##   suppressPackageStartupMessages(library(circlize))
## =====
```

```
library(ComplexHeatmap)
```

```
## Loading required package: grid

##
## Attaching package: 'grid'

## The following object is masked from 'package:Biostrings':
##
##     pattern

## =====
## ComplexHeatmap version 2.15.2
## Bioconductor page: http://bioconductor.org/packages/ComplexHeatmap/
## Github page: https://github.com/jokergoo/ComplexHeatmap
## Documentation: http://jokergoo.github.io/ComplexHeatmap-reference
##
## If you use it in published research, please cite either one:
## - Gu, Z. Complex Heatmap Visualization. iMeta 2022.
## - Gu, Z. Complex heatmaps reveal patterns and correlations in multidimensional
##   genomic data. Bioinformatics 2016.
##
##
## The new InteractiveComplexHeatmap package can directly export static
## complex heatmaps into an interactive Shiny app with zero effort. Have a try!
##
## This message can be suppressed by:
##   suppressPackageStartupMessages(library(ComplexHeatmap))
```

```
## =====
```

Data

the files imported here are created with RNAhybrid with the “RNAhybrid fig 2” script

```
Personalized_Reader <- function(lambda){
  read.table(lambda, sep = ":") %>% select(V1, V5, V6, V7, V10, V11)}

#File lists
reslistA <- list.files(path = "D:/Krueger_Lab/Publications/miR181_paper_nongithub/Figure2/RNAhybrid/res")
reslistB <- list.files(path = "D:/Krueger_Lab/Publications/miR181_paper_nongithub/Figure2/RNAhybrid/res")

#import
myfilelistA <- lapply(reslistA, Personalized_Reader)
myfilelistB <- lapply(reslistB, Personalized_Reader)

resframeA <- bind_rows(myfilelistA)
resframeB <- bind_rows(myfilelistB)

#colnames
colnames(resframeA) <- c("rownumber", "mfs", "pvalue", "start_position", "binding_bases", "non_binding_bases")
colnames(resframeB) <- c("rownumber", "mfs", "pvalue", "start_position", "binding_bases", "non_binding_bases")

resframeA[is.na(resframeA$non_binding_bases), "non_binding_bases"] <- ""
resframeB[is.na(resframeB$non_binding_bases), "non_binding_bases"] <- ""

head(resframeA)
```

```
##   rownumber   mfs   pvalue start_position
## 1         1 -13.1 1.000000           87
## 2         10 -15.7 0.999882           93
## 3        100 -19.3 0.646155           36
## 4       1000 -21.9 0.197373            4
## 5      10000 -25.4 0.026603           54
## 6     10001 -18.1 0.883059           18
##                                     binding_bases non_binding_bases
## 1                      GAGUG G GUC CAA                      U      CU G CUUACAA
## 2                      G GCUGUC                      UGA UG      GCAACUUACAA
## 3    AGU GGCUGUCG ACU                      UACAA UG      CA
## 4                      GUGG UG      UCGCAACU CA      UGA C      UA A
## 5                      UGAG GGCUG CG CAAC UUACA                      U      U      A
## 6                      UGAGUGGC UGU CG                      CAACUUACAA
```

```
head(resframeB)

##   rownumber   mfs   pvalue start_position
## 1         1 -10.9 1.000000           84
##                                     binding_bases
## 1                      UUGGG GG      GUC U
```

```
## 2      10 -23.1 0.116906      85      UUGGG  UGGC  GUCGUU  CUU
## 3      100 -24.3 0.059292     24      UGGGUGG  UG      UCG  UUACU
## 4      1000 -21.3 0.302779      5      UGGGUGG  UGU  CGU  ACU   CA
## 5      10000 -23.8 0.078879     53      UUGGG  GGCUG  CGUU  AC  UUACA
## 6      10001 -19.2 0.711209     18  UGGGUGGC  UGU  CGUU      ACUU

##              non_binding_bases
## 1              U      CU      G  UACUUACAA
## 2              U              A      ACAA
## 3      U              C              UACAA
## 4              U      C              U  UA      A
## 5              U      U              A
## 6 U              ACAA
```

Ribo profiling data

```
RNA <- read.csv("D:/Krueger_Lab/Publications/miR181_paper/Figure3/RNA_masterframe.csv")
RPF <- read.csv("D:/Krueger_Lab/Publications/miR181_paper/Figure3/RPF_masterframe.csv")
```

original bs data

```
mir181bs <- as.data.frame(readRDS("D:/Krueger_Lab/Publications/miR181_paper/Figure1/mir181_binding_sites.rds"))
mir181bs$rownumber <- 1:length(mir181bs$seqnames)
```

colours

#colours

```
farbeneg <- "#b4b4b4"
farbe1 <- "#0073C2FF"
farbe2 <- "#EFC000FF"
farbe3 <- "#CD534CFF"
farbe4 <- "#7AA6DCFF"
farbe5 <- "#868686FF"
farbe6 <- "#003C67FF"
farbe7 <- "#8F7700FF"
farbe8 <- "#3B3B3BFF"
farbe9 <- "#A73030FF"
farbe10 <- "#4A6990FF"
farbe11 <- "#FF6F00FF"
farbe12 <- "#C71000FF"
farbe13 <- "#008EAOFF"
farbe14 <- "#8A4198FF"
farbe15 <- "#5A9599FF"
farbe16 <- "#FF6348FF"

RNApcol <- "#b56504"
RNAncol <- "#027d73"
RPFpcol <- "#c4c404"
RPFncol <- "#8d0391"
```

Process data (remove gaps)

Due to the loops in the mRNA there are additional spaces in the mirna. We only want the binding and non binding bases of the mirna in the correct order. For that we will remove all gaps that originate in the mRNA loops.

```

#binding and non binding bases as characters in a list
Alistbb <- strsplit(resframeA$binding_bases,"")
Alistnb <- strsplit(resframeA$non_binding_bases,"")

Blistbb <- strsplit(resframeB$binding_bases,"")
Blistnb <- strsplit(resframeB$non_binding_bases,"")

#combine the two lists
Alist <- Map(cbind, Alistbb, Alistnb)

## Warning in cbind(...): number of rows of result is not a multiple of vector
## length (arg 2)

## Warning in cbind(...): number of rows of result is not a multiple of vector
## length (arg 2)

## Warning in cbind(...): number of rows of result is not a multiple of vector
## length (arg 2)

## Warning in cbind(...): number of rows of result is not a multiple of vector
## length (arg 2)

## Warning in cbind(...): number of rows of result is not a multiple of vector
## length (arg 2)

Alist <- lapply(Alist, as.data.frame)

Blist <- Map(cbind, Blistbb, Blistnb)

## Warning in cbind(...): number of rows of result is not a multiple of vector
## length (arg 2)

## Warning in cbind(...): number of rows of result is not a multiple of vector
## length (arg 2)

## Warning in cbind(...): number of rows of result is not a multiple of vector
## length (arg 2)

## Warning in cbind(...): number of rows of result is not a multiple of vector
## length (arg 2)

## Warning in cbind(...): number of rows of result is not a multiple of vector
## length (arg 2)

## Warning in cbind(...): number of rows of result is not a multiple of vector
## length (arg 2)

```

```
## Warning in cbind(...): number of rows of result is not a multiple of vector
## length (arg 2)
```

```
## Warning in cbind(...): number of rows of result is not a multiple of vector
## length (arg 2)
```

```
## Warning in cbind(...): number of rows of result is not a multiple of vector
## length (arg 2)
```

```
## Warning in cbind(...): number of rows of result is not a multiple of vector
## length (arg 2)
```

```
Blist <- lapply(Blist, as.data.frame)
```

```
#remove all empty rows (mRNA loops)
Alist0 <- lapply(Alist, function(x){
  x[!(x[,1]== " " & x[,2] == " "),]
})
```

```
Blist0 <- lapply(Blist, function(x){
  x[!(x[,1]== " " & x[,2] == " "),]
})
```

```
#rewrite as characters
AlistF <- lapply(Alist0, function(x){
  paste(x[,1], collapse = '')
})
```

```
BlistF <- lapply(Blist0, function(x){
  paste(x[,1], collapse = '')
})
```

```
#Attach lists back onto original data.frame as new column
resframeA$binding_nospace <-unlist(AlistF)
head(resframeA$binding_nospace)
```

```
## [1] " GAGUGG  GUC CAA      " "  G  GCUGUC      "
## [3] "  AGUGGCUGUCG  ACUUACAA" "   GUGG  UGUCGCAACU  CA "
## [5] "UGAG  GGCUG  CGCAACUUACA " "UGAGUGGCUGUCG      "
```

```
resframeB$binding_nospace <-unlist(BlistF)
head(resframeB$binding_nospace)
```

```
## [1] "UUGGG GG GUC U      " "UUGGGUGGC GUCGUU CUU  "
## [3] " UGGGUGG UGUCGUUACU    " " UGGGUGG UGUCGU ACU  CA "
## [5] "UUGGG GGCUG CGUUACUUACA " " UGGGUGGCUGUCGUUACUU  "
```

Transform into Numbers

add 0s

replace all gaps with 0 and all letters with 1

```
#0
```

```
resframeA$binding_nospace <- chartr(" ", "0", resframeA$binding_nospace)
resframeB$binding_nospace <- chartr(" ", "0", resframeB$binding_nospace)
```

```
#1
```

```
resframeA$binding_nospace <- mgsub::mgsub(resframeA$binding_nospace, c("A", "U", "C", "G"), c(rep("1", 4), rep("0", 16)))
resframeB$binding_nospace <- mgsub::mgsub(resframeB$binding_nospace, c("A", "U", "C", "G"), c(rep("1", 4), rep("0", 16)))
```

```
head(resframeA)
```

```
##   rownumber   mfs   pvalue start_position
## 1         1 -13.1 1.000000             87
## 2        10 -15.7 0.999882             93
## 3       100 -19.3 0.646155             36
## 4      1000 -21.9 0.197373              4
## 5     10000 -25.4 0.026603             54
## 6    10001 -18.1 0.883059             18
##                                     binding_bases
## 1                GAGUG G GUC CAA
## 2                G GCUGUC
## 3      AGU GGCUGUCG ACU          UACAA UG
## 4                GUGG UG   UCGCAACU CA
## 5                UGAG GGCUG CG CAAC UUACA
## 6                UGAGUGGC UGU CG
##                                     non_binding_bases
## 1                U          CU G CUUACAA
## 2                UGA UG          GCAACUUACAA
## 3                CA
## 4      UGA      C          UA  A
## 5                U      U          A
## 6                CAACUUACAA
##                                     binding_nospace
## 1 011111100111011100000000
## 2 000100111111100000000000
## 3 001111111111100111111111
## 4 000111101111111111100110
## 5 111101111101111111111110
## 6 111111111111100000000000
```

```
head(resframeB)
```

```
##   rownumber   mfs   pvalue start_position
## 1         1 -10.9 1.000000             84
## 2        10 -23.1 0.116906             85
## 3       100 -24.3 0.059292             24
## 4      1000 -21.3 0.302779              5
##                                     binding_bases
## 1      UUGGG GG GUC U
## 2      UUGGG UGGC GUCGUU CUU
## 3      UGGGUGG UG   UCG UUACU
## 4      UGGGUGG UGU CGU ACU  CA
```



```
## 5      10000 -23.8 0.078879          53      UUGGG GGCUG CGUU AC UUACA
## 6      10001 -19.2 0.711209          18      UGGGUGGC UGU CGUU      ACUU
##              non_binding_bases      binding_nospace
## 1              U      CU      G UACUUACAA 111110110011101000000000
## 2              U              A      ACAA 111111111011111101110000
## 3      U              C              UACAA 011111110111111111100000
## 4              U      C              U      UA      A 011111110111111011100110
## 5              U      U              A 111110111110111111111110
## 6 U              ACAA 011111111111111111111110000
```

seperate into columns

for each base make 1 column so it can be added and also put into a heatmap

```
#for the heatmap with every binding site
heatframeA <- do.call(rbind.data.frame, strsplit(resframeA$binding_nospace,""))
heatframeA <- sapply( heatframeA, as.numeric )
colnames(heatframeA) <- c(23:1)
rownames(heatframeA) <- resframeA[,1]
head(heatframeA)
```

```
##      23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
## 1      0 1 1 1 1 1 1 0 0 1 1 1 0 1 1 1 0 0 0 0 0 0
## 10     0 0 0 1 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
## 100    0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1
## 1000   0 0 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1
## 10000  1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0
## 10001  1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
```

```
heatframeB <- do.call(rbind.data.frame, strsplit(resframeB$binding_nospace,""))
heatframeB <- sapply( heatframeB, as.numeric )
colnames(heatframeB) <- c(24:1)
rownames(heatframeB) <- resframeB[,1]
head(heatframeB)
```

```
##      24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
## 1      1 1 1 1 1 0 1 1 0 0 1 1 1 0 1 0 0 0 0 0 0 0
## 10     1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 0 0
## 100    0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 0
## 1000   0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 0 0 1
## 10000  1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0
## 10001  0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
```

```
#reverse column order
heatframeA <-heatframeA[,23:1]
heatframeB <- heatframeB[,24:1]
```

sum of columns

```
#sum for the small heatmap with the overall binding ratio for each base
framesumA <- colSums(heatframeA)
framesumB <- colSums(heatframeB)

framesum <- as.data.frame(rbind(framesumA,framesumB))
```

```
## Warning in rbind(...): number of columns of result is not a multiple of vector
## length (arg 1)
```

```
rownames(framesum) <- c("miR181a", "miR181b")
framesum
```

```
##           1      2      3      4      5      6      7      8      9     10     11     12     13
## miR181a 1205  4404  6851  6918  7960  8676   9500  7229  5796  7656 11289 12224 11736
## miR181b 1440  4912  7353  7449  8344  9403 10549  8757  9050  9925 11709 12121 11389
##           14      15      16      17      18      19      20      21      22      23      24
## miR181a 11082 10649 10034 11382 11281 11109 11391  9738  9347  6328 1205
## miR181b 10761 10522 10129 11123 11045 11178 12424 12436 11414  8803  6027
```

```
#scale for better comperativity
sframesum <- as.data.frame(t(scale(t(framesum))))
```

Heatmap

Colours

```
hmcols1 <- c("white", "black")
hmcols2 <- colorRamp2(c(-2, 2), c("white", "red"))
```

Heatmap of all the single reads

make heatmap without column clustering but with row clustering

```
HMA <- Heatmap(heatframeA, cluster_columns = F, col = hmcols1, row_km = 5, show_row_names = F, show_row
```

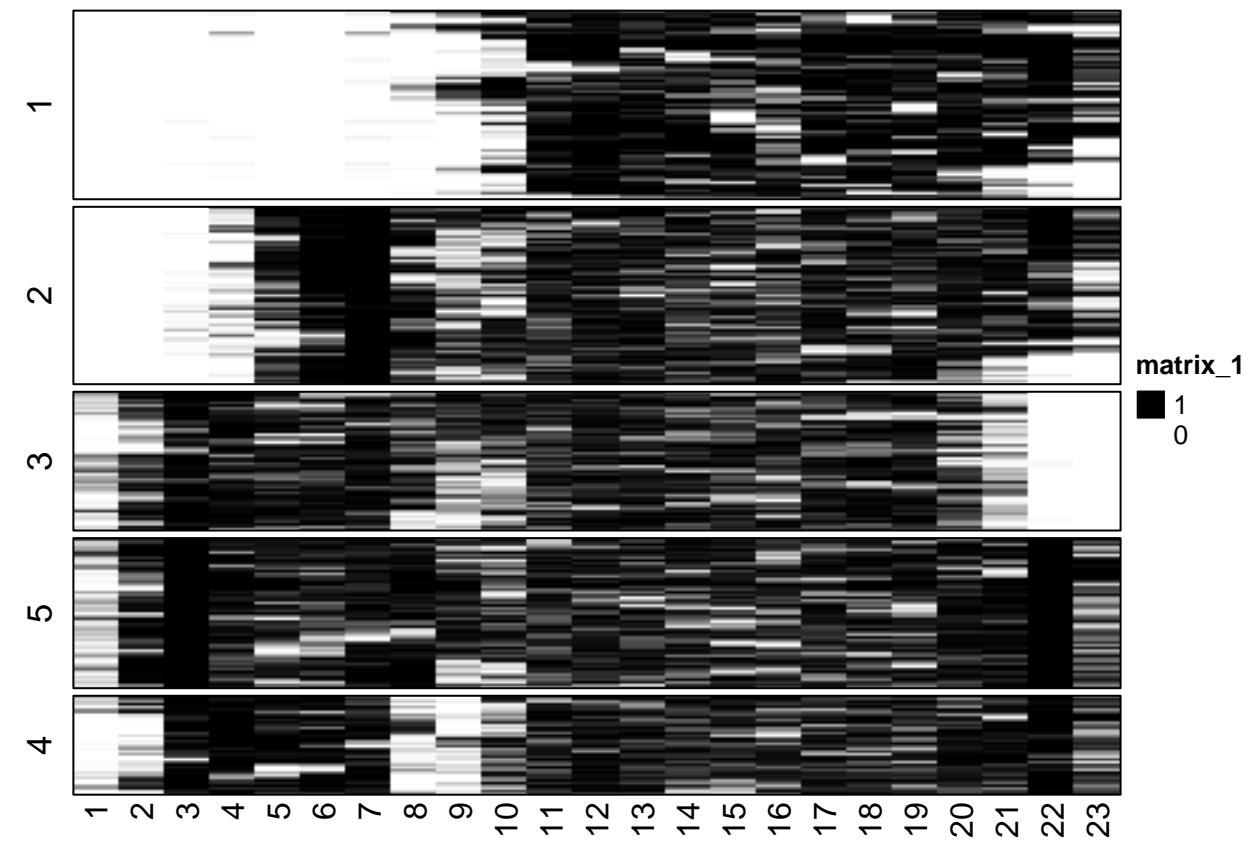
```
## `use_raster` is automatically set to TRUE for a matrix with more than
## 2000 rows. You can control `use_raster` argument by explicitly setting
## TRUE/FALSE to it.
##
## Set `ht_opt$message = FALSE` to turn off this message.
```

```
HMB <- Heatmap(heatframeB, cluster_columns = F, col = hmcols1, row_km = 5, show_row_names = F, show_row
```

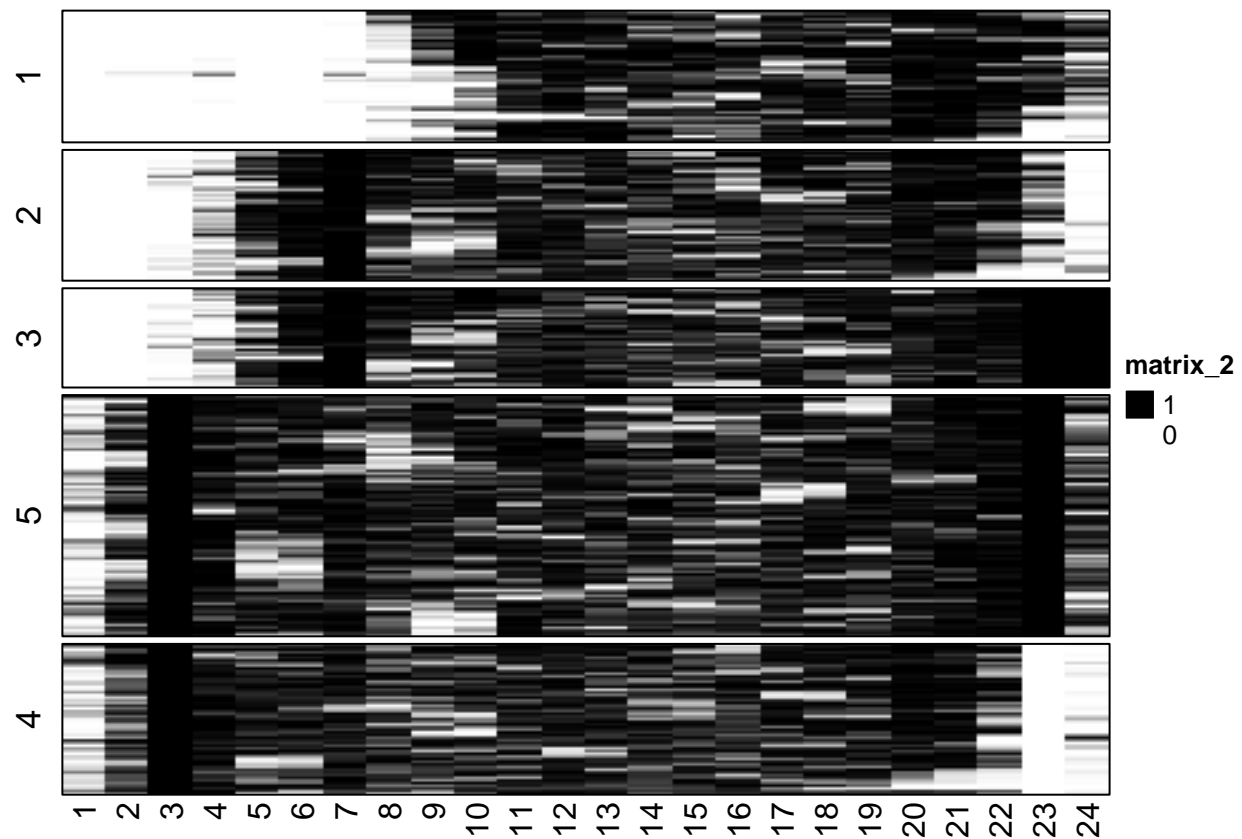
```
## `use_raster` is automatically set to TRUE for a matrix with more than
## 2000 rows. You can control `use_raster` argument by explicitly setting
## TRUE/FALSE to it.
##
## Set `ht_opt$message = FALSE` to turn off this message.
```

No B

```
HMA
```



B
HMB



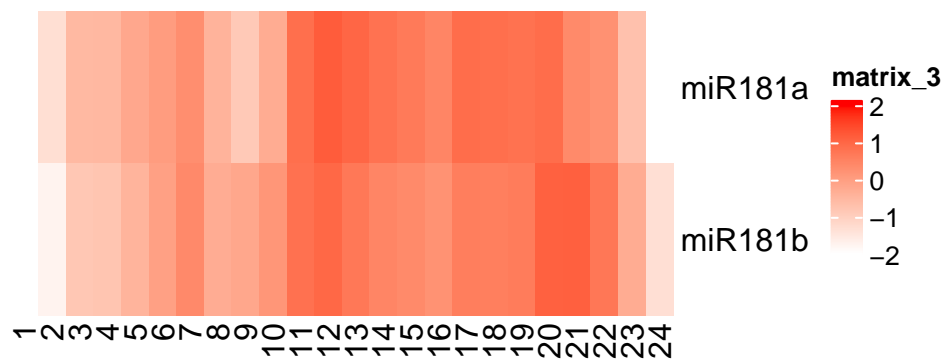
“Heatmap” of combined reads for mir_181a and b

No clustering, only sums

```
HMF <- Heatmap(sframesum, cluster_columns = F, cluster_rows = F, col = hmcols2)
```

Warning: The input is a data frame-like object, convert it to a matrix.

```
HMF
```



cluster seperately

clustering

try to cluster seperately

```
#cluster by seed area
heat_ksA <- kmeans(heatframeA, centers = 5)
heat_k_namesA <- as.data.frame(heat_ksA$cluster)
#merge back with full data and adjust frame again
cframeA <- merge(heatframeA, heat_k_namesA, by=0)
rownames(cframeA) <- cframeA$Row.names
cframeA <- cframeA[,-1]

#order by clusters (will be needed for heatmap without clustering)
cframeA <- cframeA[order(cframeA$`heat_ksA$cluster`, decreasing = F),]
#remove cluster col
cframeAp <- cframeA[,-24]
head(cframeAp)
```

```
##          1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
## 10000 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1
## 10003 0 0 1 1 1 1 0 0 0 1 1 1 1 1 0 0 1 1 1 1 1 1 0
## 10011 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## 10012 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 0
## 10016 0 0 1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1
## 10017 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 0 0 1 0 1 1 1
```

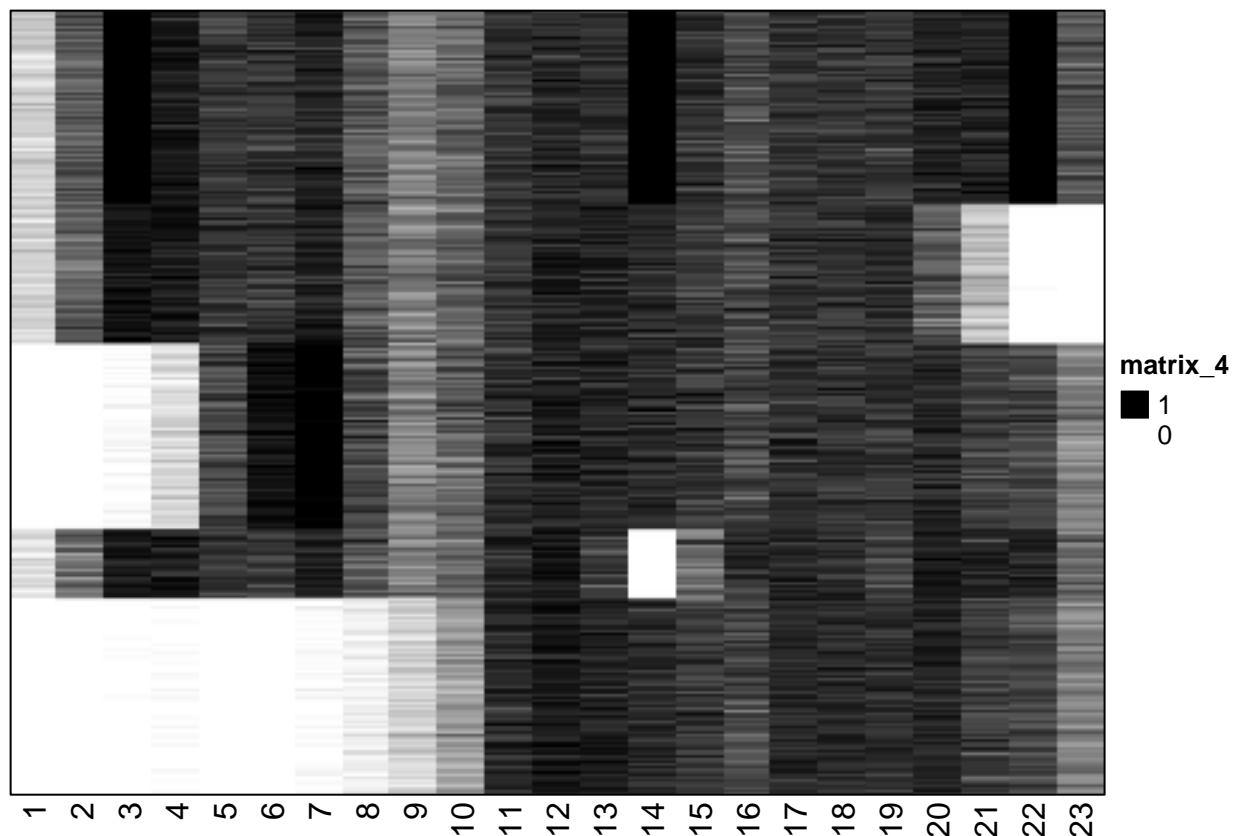
plot clustered seperately

```
HMAsep <- Heatmap(cframeAp, cluster_columns = F, cluster_rows = F, col = hmcols1, show_row_names = F, s

## Warning: The input is a data frame-like object, convert it to a matrix.

## `use_raster` is automatically set to TRUE for a matrix with more than
## 2000 rows. You can control `use_raster` argument by explicitly setting
## TRUE/FALSE to it.
##
## Set `ht_opt$message = FALSE` to turn off this message.

HMAsep
```



clustering by seed region

try to cluster separately only by the binding bases

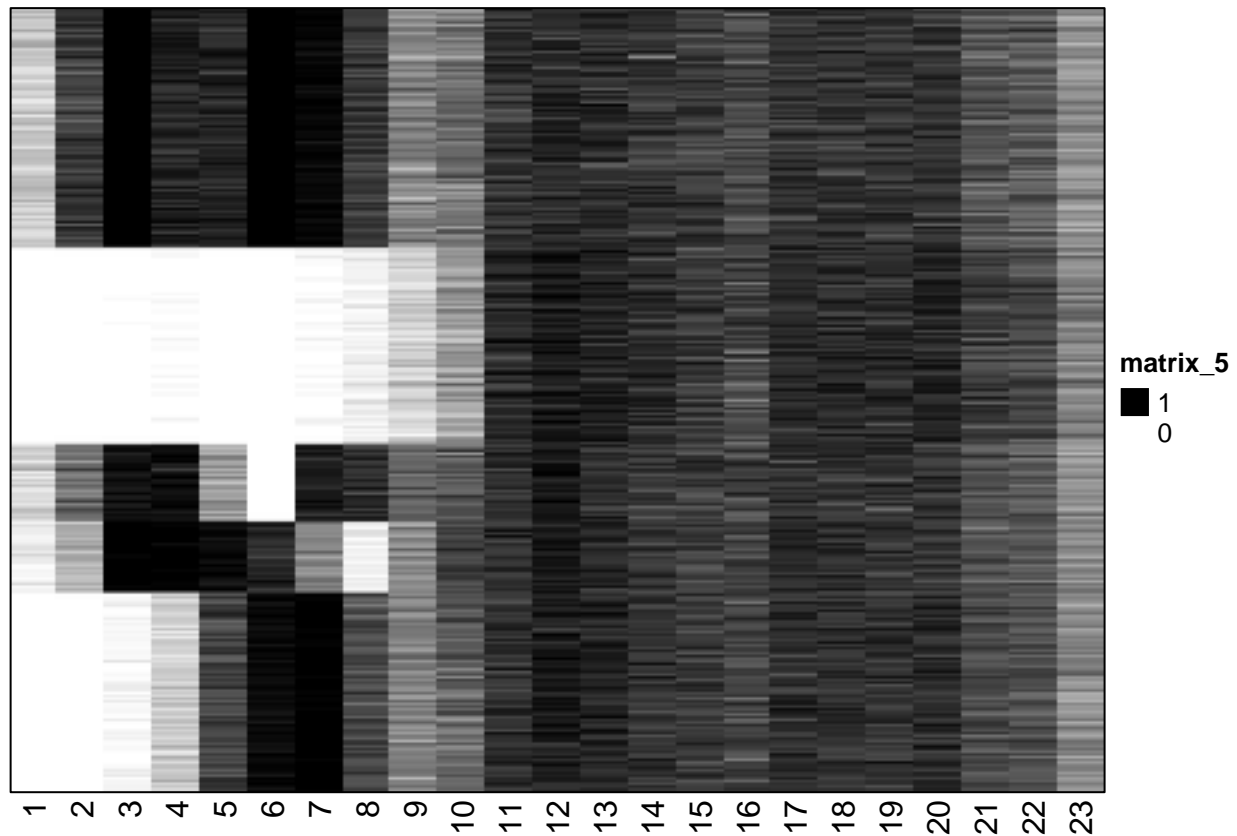
```
#cluster by seed area
heat_ksAseed <- kmeans(heatframeA[,1:8], centers = 5)
heat_k_namesAseed <- as.data.frame(heat_ksAseed$cluster)
#merge back with full data and adjust frame again
cframeAseed <- merge(heatframeA, heat_k_namesAseed, by=0)
rownames(cframeAseed) <- cframeAseed$Row.names
cframeAseed <- cframeAseed[,-1]

#order by clusters (will be needed for heatmap without clustering)
cframeAseed <- cframeAseed[order(cframeAseed$`heat_ksAseed$cluster`, decreasing = F),]
#remove cluster col
cframeAseedp <- cframeAseed[,-24]
head(cframeAseedp)
```

```
##      1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
## 100   1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0
## 1000  0 1 1 0 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 0
## 10000 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1
## 10009 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 0 0 0
## 10010 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0
## 10015 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0
```

plot clustered by seed region

```
HMAseed <- Heatmap(cframeAseedp, cluster_columns = F, cluster_rows = F, col = hmcols1, show_row_names =  
## Warning: The input is a data frame-like object, convert it to a matrix.  
## `use_raster` is automatically set to TRUE for a matrix with more than  
## 2000 rows. You can control `use_raster` argument by explicitly setting  
## TRUE/FALSE to it.  
##  
## Set `ht_opt$message = FALSE` to turn off this message.  
HMAseed
```



testcode

```
distframe <- dist(heatframeA) head(distframe) clustobj <- hclust(distframe)  
plot(clustobj)
```

ECDF plots

merge back with original data for gene names

```
cframeA$rownumber <- as.numeric(rownames(cframeA))
```

```
bsseqHA <- left_join(mir181bs, cframeA, by="rownumber")
```

```
head(bsseqHA)
```

```
##      seqnames      start      end width strand scoreSum scoreMean scoreMax
## 1      chr1 6245651 6245657      7      + 9.52553 4.762765 6.00678
## 2      chr1 6248341 6248347      7      + 92.68921 23.172303 48.76900
## 3      chr1 6248857 6248863      7      + 14.07133 7.035665 7.04425
## 4      chr1 6248918 6248924      7      + 38.91451 12.971503 20.65080
## 5      chr1 7170481 7170487      7      + 66.92218 13.384436 25.84490
## 6      chr1 9899605 9899611      7      + 25.15963 6.289907 8.61019
##      geneType geneName      geneID region BS_ID      mir_IP
## 1 protein_coding Rb1cc1 ENSMUSG000000025907 cds      5 mmu-miR-181a-5p
## 2 protein_coding Rb1cc1 ENSMUSG000000025907 cds      8 mmu-miR-181a-5p
## 3 protein_coding Rb1cc1 ENSMUSG000000025907 cds     10 mmu-miR-181a-5p
## 4 protein_coding Rb1cc1 ENSMUSG000000025907 cds     11 mmu-miR-181a-5p
## 5 protein_coding Pcmt1 ENSMUSG000000051285 utr3     19 mmu-miR-181a-5p
## 6 protein_coding Sgk3 ENSMUSG000000025915 utr3     23 mmu-miR-181a-5p
##      n_mir181 n_mir181a n_mir181b n_mir181c n_mir181d      set WT KO
## 1          1          1          0          0          0 ago_bs_mir181_chi 1 1
## 2          5          5          0          0          0 ago_bs_mir181_chi 1 1
## 3          6          6          0          0          0 ago_bs_mir181_chi 1 0
## 4          6          6          0          0          0 ago_bs_mir181_chi 1 1
## 5          4          4          0          0          0 ago_bs_mir181_chi 1 1
## 6          1          1          0          0          0 ago_bs_mir181_chi NA NA
##      geneID.2 geneName.1 region.1 counts.bs.1_KO counts.bs.2_KO
## 1 ENSMUSG000000025907 Rb1cc1 cds          4          3
## 2 ENSMUSG000000025907 Rb1cc1 cds         28         32
## 3 ENSMUSG000000025907 Rb1cc1 cds         13         11
## 4 ENSMUSG000000025907 Rb1cc1 cds         15         15
## 5 ENSMUSG000000051285 Pcmt1 utr3         12         22
## 6          <NA>          <NA>          <NA>          NA          NA
##      counts.bs.3_KO counts.bs.4_WT counts.bs.5_WT counts.bs.6_WT
## 1          3          3          10          3
## 2         27         46          41         20
## 3          4         22          13         12
## 4         10         33          20         18
## 5         14         16          20          9
## 6         NA         NA          NA         NA
##      geneID.1 counts.bg.1_KO counts.bg.2_KO counts.bg.3_KO
## 1 ENSMUSG000000025907      1609      1973      1250
## 2 ENSMUSG000000025907      1609      1973      1250
## 3 ENSMUSG000000025907      1609      1973      1250
## 4 ENSMUSG000000025907      1609      1973      1250
## 5 ENSMUSG000000051285     1355     1706     1064
## 6          <NA>          NA          NA          NA
##      counts.bg.4_WT counts.bg.5_WT counts.bg.6_WT resBs.baseMean
## 1         2638         2231         1352         92.10645
## 2         2638         2231         1352        281.53271
## 3         2638         2231         1352        145.51107
## 4         2638         2231         1352        186.74162
## 5         1654         1348          755        151.36245
## 6          NA          NA          NA          NA
##      resBs.log2FoldChange resBs.lfcSE resBs.stat resBs.pvalue resBs.padj
```



```

## 1      -0.1093039    0.5923673 0.03419066    0.8533018 0.9652601
## 2      0.2749428    0.2351157 1.35874137    0.2437557 0.6729889
## 3     -0.1805519    0.3623758 0.25017050    0.6169550 0.8961239
## 4     -0.2606282    0.3062717 0.73169661    0.3923338 0.7868678
## 5      0.1466485    0.3122905 0.22052922    0.6386370 0.9013566
## 6      NA          NA          NA          NA          NA
##      resBg.baseMean resBg.log2FoldChange resBg.lfcSE resBg.stat resBg.pvalue
## 1      NA          NA          NA          NA          NA
## 2      NA          NA          NA          NA          NA
## 3      NA          NA          NA          NA          NA
## 4      NA          NA          NA          NA          NA
## 5      NA          NA          NA          NA          NA
## 6      NA          NA          NA          NA          NA
##      resBg.padj tpm.counts.bg.1_KO tpm.counts.bg.2_KO tpm.counts.bg.3_KO
## 1      NA          133.7259          117.9980          129.8669
## 2      NA          133.7259          117.9980          129.8669
## 3      NA          133.7259          117.9980          129.8669
## 4      NA          133.7259          117.9980          129.8669
## 5      NA          248.6210          225.2505          244.0445
## 6      NA          NA          NA          NA
##      tpm.counts.bg.4_WT tpm.counts.bg.5_WT tpm.counts.bg.6_WT
## 1      139.8635          146.2855          163.5360
## 2      139.8635          146.2855          163.5360
## 3      139.8635          146.2855          163.5360
## 4      139.8635          146.2855          163.5360
## 5      193.5994          195.1330          201.6149
## 6      NA          NA          NA
##      BS_ID.1 tpm_support_KO tpm_support_WT tpm_supported down
## 1 ENSMUSG00000025907.bs5          3          3          TRUE FALSE
## 2 ENSMUSG00000025907.bs8          3          3          TRUE FALSE
## 3 ENSMUSG00000025907.bs10         3          3          TRUE FALSE
## 4 ENSMUSG00000025907.bs11         3          3          TRUE FALSE
## 5 ENSMUSG00000051285.bs4          3          3          TRUE FALSE
## 6      <NA>          NA          NA          NA      NA
##      rownumber 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
## 1      1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 0 0 1 1 1 1 1 0
## 2      2 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
## 3      3 0 0 0 0 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1
## 4      4 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 0 1 1 1 1
## 5      5 1 1 1 1 1 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0
## 6      6 0 1 1 1 1 1 0 1 1 1 0 1 1 0 1 1 1 0 0 1 1 1 1
##      heat_ksA$cluster
## 1      5
## 2      5
## 3      3
## 4      1
## 5      2
## 6      4

```

sort RNA and RPF by cluster

```

#RNA
colnames(RNA)[16] <- "geneName"
RNA <- left_join(RNA, bsseqHA[!duplicated(bsseqHA$geneName)], by="geneName")
RNA[is.na(RNA$heat_ksA$cluster), "heat_ksA$cluster"] <- "Non-target"

```

```
head(RNA)
```

```
##      X      Gene      baseMean log2FoldChange      lfcSE      stat
## 1 1  ENSMUSG000000104197.1 2009.89774      2.741830 0.10245760 26.76063
## 2 2  ENSMUSG000000004110.17 1028.70696      -2.019241 0.09914247 -20.36706
## 3 3  ENSMUSG000000027669.14  648.18701      1.905815 0.10065041 18.93499
## 4 4  ENSMUSG000000098206.1 4004.62856      2.148076 0.12889738 16.66501
## 5 5  ENSMUSG000000004552.16  516.71005      -2.252195 0.13648248 -16.50172
## 6 6  ENSMUSG000000069306.5   68.07176      2.763638 0.19108029 14.46323
##      pvalue      padj      WT_1411      WT_1601      WT_1710      KO_1411
## 1 9.288051e-158 1.183948e-153 487.52165 474.5874 405.8464624 3338.7077
## 2 3.277594e-92 2.088974e-88 1576.25174 1737.4574 1739.7225816 400.4560
## 3 5.872559e-80 2.495250e-76 248.88185 252.5740 252.2109307 1048.3637
## 4 2.354521e-62 7.503269e-59 1449.25030 1259.2746 985.7539896 5751.8330
## 5 3.566111e-61 9.091443e-58 872.62279 1026.4751 777.9463918 148.2821
## 6 2.068594e-47 4.394729e-44 2.04841 0.0000 0.8880667 123.7258
##      KO_1601      KO_1710      LFCandPADJSig      geneName      seqnames      start
## 1 3428.5263 3924.1970      Significant up      Gm37632      <NA>      NA
## 2 357.0428 361.3113      Significant down      Cacna1e      chr1 154633673
## 3 1043.6635 1043.4281      Significant up      Gnb4      <NA>      NA
## 4 6644.9630 7936.6965      Significant up A430106G13Rik <NA>      NA
## 5 112.9110 162.0230      Significant down      Ctse      chr1 131672503
## 6 146.4791 135.2892      Significant up      Hist1h4m      <NA>      NA
##      end width strand scoreSum scoreMean scoreMax      geneType
## 1      NA      NA      <NA>      NA      NA      NA      <NA>
## 2 154633679      7      - 10.09826 5.04913 8.4785 protein_coding
## 3      NA      NA      <NA>      NA      NA      NA      <NA>
## 4      NA      NA      <NA>      NA      NA      NA      <NA>
## 5 131672509      7      + 47.26199 11.81550 19.3220 protein_coding
## 6      NA      NA      <NA>      NA      NA      NA      <NA>
##      geneID region BS_ID      mir_IP n_mir181 n_mir181a n_mir181b
## 1      <NA>      <NA>      <NA>      <NA>      NA      NA      NA
## 2 ENSMUSG000000004110      utr5      <NA>      <NA>      NA      NA      NA
## 3      <NA>      <NA>      <NA>      <NA>      NA      NA      NA
## 4      <NA>      <NA>      <NA>      <NA>      NA      NA      NA
## 5 ENSMUSG000000004552      cds      382 mmu-miR-181a-5p      5      2      3
## 6      <NA>      <NA>      <NA>      <NA>      NA      NA      NA
##      n_mir181c n_mir181d      set WT KO      geneID.2 geneName.1
## 1      NA      NA      <NA> NA NA      <NA>      <NA>
## 2      NA      NA      mir181_enriched NA NA      <NA>      <NA>
## 3      NA      NA      <NA> NA NA      <NA>      <NA>
## 4      NA      NA      <NA> NA NA      <NA>      <NA>
## 5      0      0 ago_bs_mir181_chi 1 0 ENSMUSG000000004552      Ctse
## 6      NA      NA      <NA> NA NA      <NA>      <NA>
##      region.1 counts.bs.1_KO counts.bs.2_KO counts.bs.3_KO counts.bs.4_WT
## 1      <NA>      NA      NA      NA      NA
## 2      <NA>      NA      NA      NA      NA
## 3      <NA>      NA      NA      NA      NA
## 4      <NA>      NA      NA      NA      NA
## 5      cds      0      0      1      15
## 6      <NA>      NA      NA      NA      NA
##      counts.bs.5_WT counts.bs.6_WT      geneID.1 counts.bg.1_KO
## 1      NA      NA      <NA>      NA
## 2      NA      NA      <NA>      NA
```

```

## 3          NA          NA          <NA>          NA
## 4          NA          NA          <NA>          NA
## 5          9          6 ENSMUSG00000004552          431
## 6          NA          NA          <NA>          NA
## counts.bg.2_KO counts.bg.3_KO counts.bg.4_WT counts.bg.5_WT counts.bg.6_WT
## 1          NA          NA          NA          NA          NA
## 2          NA          NA          NA          NA          NA
## 3          NA          NA          NA          NA          NA
## 4          NA          NA          NA          NA          NA
## 5          735          417          1560          1199          558
## 6          NA          NA          NA          NA          NA
## resBs.baseMean resBs.log2FoldChange resBs.lfcSE resBs.stat resBs.pvalue
## 1          NA          NA          NA          NA          NA
## 2          NA          NA          NA          NA          NA
## 3          NA          NA          NA          NA          NA
## 4          NA          NA          NA          NA          NA
## 5          57.03841          -3.535302          1.220067          13.8548          0.000197492
## 6          NA          NA          NA          NA          NA
## resBs.padj resBg.baseMean resBg.log2FoldChange resBg.lfcSE resBg.stat
## 1          NA          NA          NA          NA          NA
## 2          NA          NA          NA          NA          NA
## 3          NA          NA          NA          NA          NA
## 4          NA          NA          NA          NA          NA
## 5          0.009740614          NA          NA          NA          NA
## 6          NA          NA          NA          NA          NA
## resBg.pvalue resBg.padj tpm.counts.bg.1_KO tpm.counts.bg.2_KO
## 1          NA          NA          NA          NA
## 2          NA          NA          NA          NA
## 3          NA          NA          NA          NA
## 4          NA          NA          NA          NA
## 5          NA          NA          165.9967          203.7031
## 6          NA          NA          NA          NA
## tpm.counts.bg.3_KO tpm.counts.bg.4_WT tpm.counts.bg.5_WT tpm.counts.bg.6_WT
## 1          NA          NA          NA          NA
## 2          NA          NA          NA          NA
## 3          NA          NA          NA          NA
## 4          NA          NA          NA          NA
## 5          200.7646          383.2804          364.3204          312.776
## 6          NA          NA          NA          NA
## BS_ID.1 tpm_support_KO tpm_support_WT tpm_supported down
## 1          <NA>          NA          NA          NA          NA
## 2          <NA>          NA          NA          NA          NA
## 3          <NA>          NA          NA          NA          NA
## 4          <NA>          NA          NA          NA          NA
## 5 ENSMUSG00000004552.bs4          3          3          TRUE TRUE
## 6          <NA>          NA          NA          NA          NA
## rownumber 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
## 1          NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
## 2          7405 0 0 1 1 0 0 1 1 1 1 1 1 1 1 1 0 1 0 1 1 0
## 3          NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
## 4          NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
## 5          103 1 1 1 1 0 1 1 0 0 0 1 1 1 1 0 1 1 1 0 1 0
## 6          NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
## 23 heat_ksA$cluster

```

```
## 1 NA      Non-target
## 2 0        2
## 3 NA      Non-target
## 4 NA      Non-target
## 5 0        2
## 6 NA      Non-target
```

```
table(RNA$`heat_ksA$cluster`)
```

```
##
##          1          2          3          4          5 Non-target
##        843        634        866        302        899        9757
```

```
#RPF
```

```
colnames(RPF)[16] <- "geneName"
RPF <- left_join(RPF, bsseqHA[!duplicated(bsseqHA$geneName),], by="geneName")
RPF[is.na(RPF$`heat_ksA$cluster`), "heat_ksA$cluster"] <- "Non-target"
head(RPF)
```

```
## X          Gene baseMean log2FoldChange lfcSE      stat
## 1 1 ENSMUSG000000000001.4 3788.57802    0.05462417 0.05957247 0.9169365
## 2 2 ENSMUSG000000000028.15 1086.36203   -0.12956555 0.07692175 -1.6843811
## 3 3 ENSMUSG000000000037.17  19.45454   -0.06042604 0.15784031 -0.3828302
## 4 4 ENSMUSG000000000056.7 1454.64304    0.05456272 0.08781406 0.6213438
## 5 5 ENSMUSG000000000078.7  850.27809    0.65975317 0.07971920 8.2759636
## 6 6 ENSMUSG000000000085.16 337.25837   -0.06342545 0.12383411 -0.5121808
##      pvalue      padj    WT_1411    WT_1601    WT_1710    KO_1411
## 1 3.591759e-01 6.850504e-01 3687.40632 3612.48756 3862.10415 3905.91865
## 2 9.210804e-02 3.386014e-01 1091.79093 1147.15325 1182.54053 1081.59050
## 3 7.018457e-01 8.852986e-01  13.71269  22.17027  29.07887  28.37993
## 4 5.343734e-01 8.038393e-01 1561.28714 1359.98786 1348.39777 1344.36759
## 5 1.274210e-16 2.314117e-14  649.06709  668.27536  622.50312 1054.26168
## 6 6.085245e-01 8.452822e-01  325.18653  326.21972  396.33417  351.07019
##      KO_1601    KO_1710  LFCandPADJSig geneName seqnames      start      end
## 1 3984.05486 3679.496580 Not significant  Gnai3      chr3 108118439 108118445
## 2 1030.01306 985.083926 Not significant   Cdc45      <NA>      NA      NA
## 3  17.83572   5.549769 Not significant   Scml2     chrX 161199514 161199520
## 4 1559.88270 1553.935207 Not significant   Narf     chr11 121237269 121237275
## 5 1022.58151 1084.979761 Significant up    Klf6     chr13  5867643  5867649
## 6  350.02608 274.713546 Not significant   Scmh1     <NA>      NA      NA
##      width strand scoreSum scoreMean scoreMax      geneType      geneID
## 1      7      -  17.16656   4.29164   6.24306 protein_coding ENSMUSG000000000001
## 2     NA <NA>      NA      NA      NA      <NA>      <NA>
## 3      7      + 409.94460 136.64820 268.27800 protein_coding ENSMUSG000000000037
## 4      7      +  54.16402  10.83280  20.82290 protein_coding ENSMUSG000000000056
## 5      7      +  31.50993  10.50331  12.38550 protein_coding ENSMUSG000000000078
## 6     NA <NA>      NA      NA      NA      <NA>      <NA>
##      region BS_ID      mir_IP n_mir181 n_mir181a n_mir181b n_mir181c n_mir181d
## 1     cds 5077 mmu-miR-181a-5p      2      2      0      0      0
## 2     <NA> <NA>      <NA>      NA      NA      NA      NA      NA
## 3  intron <NA>      <NA>      NA      NA      NA      NA      NA
## 4     utr5 16933 mmu-miR-181b-5p      1      0      1      0      0
## 5     utr3 19275 mmu-miR-181a-5p      3      3      0      0      0
## 6     <NA> <NA>      <NA>      NA      NA      NA      NA      NA
##      set WT KO      geneID.2 geneName.1 region.1 counts.bs.1_KO
## 1 ago_bs_mir181_chi 1 0 ENSMUSG000000000001      Gnai3      cds      2
```

## 2	<NA>	NA	NA	<NA>	<NA>	<NA>	NA
## 3	mir181_enriched	NA	NA	<NA>	<NA>	<NA>	NA
## 4	ago_bs_mir181_chi	1	0	ENSMUSG00000000056	Narf	utr5	9
## 5	ago_bs_mir181_chi	1	1	ENSMUSG00000000078	Klf6	utr3	4
## 6	<NA>	NA	NA	<NA>	<NA>	<NA>	NA
##	counts.bs.2_KO	counts.bs.3_KO	counts.bs.4_WT	counts.bs.5_WT	counts.bs.6_WT		
## 1	5	1	9	4	0		
## 2	NA	NA	NA	NA	NA		
## 3	NA	NA	NA	NA	NA		
## 4	10	3	33	24	17		
## 5	1	4	11	7	5		
## 6	NA	NA	NA	NA	NA		
##	geneID.1	counts.bg.1_KO	counts.bg.2_KO	counts.bg.3_KO			
## 1	ENSMUSG00000000001	781	932	606			
## 2	<NA>	NA	NA	NA			
## 3	<NA>	NA	NA	NA			
## 4	ENSMUSG00000000056	747	973	503			
## 5	ENSMUSG00000000078	319	419	249			
## 6	<NA>	NA	NA	NA			
##	counts.bg.4_WT	counts.bg.5_WT	counts.bg.6_WT	resBs.baseMean			
## 1	1137	948	500	47.01432			
## 2	NA	NA	NA	NA			
## 3	NA	NA	NA	NA			
## 4	1110	1035	543	128.71190			
## 5	607	466	244	48.53992			
## 6	NA	NA	NA	NA			
##	resBs.log2FoldChange	resBs.lfcSE	resBs.stat	resBs.pvalue	resBs.padj		
## 1	-0.3611064	0.6582148	0.3064341	0.5798766299	0.88338480		
## 2	NA	NA	NA	NA	NA		
## 3	NA	NA	NA	NA	NA		
## 4	-1.2675269	0.3692805	13.1695749	0.0002845313	0.01292422		
## 5	-0.7513615	0.5789425	1.7903402	0.1808847223	0.60039272		
## 6	NA	NA	NA	NA	NA		
##	resBg.baseMean	resBg.log2FoldChange	resBg.lfcSE	resBg.stat	resBg.pvalue		
## 1	NA	NA	NA	NA	NA		
## 2	NA	NA	NA	NA	NA		
## 3	NA	NA	NA	NA	NA		
## 4	783.0873	-0.1165693	0.09866149	-1.181507	0.23740125		
## 5	363.8138	-0.2226921	0.10871231	-2.048453	0.04051564		
## 6	NA	NA	NA	NA	NA		
##	resBg.padj	tpm.counts.bg.1_KO	tpm.counts.bg.2_KO	tpm.counts.bg.3_KO			
## 1	NA	292.31323	251.01601	283.52995			
## 2	NA	NA	NA	NA			
## 3	NA	NA	NA	NA			
## 4	0.4626108	189.76592	177.86829	159.73294			
## 5	0.1364387	88.69694	83.83397	86.54584			
## 6	NA	NA	NA	NA			
##	tpm.counts.bg.4_WT	tpm.counts.bg.5_WT	tpm.counts.bg.6_WT				
## 1	271.4737	279.9290	272.36080				
## 2	NA	NA	NA				
## 3	NA	NA	NA				
## 4	179.8832	207.4341	200.75881				
## 5	107.6655	102.2224	98.73814				
## 6	NA	NA	NA				

```
##          BS_ID.1 tpm_support_KO tpm_support_WT tpm_supported down
## 1 ENSMUSG00000000001.bs2          3          3      TRUE FALSE
## 2          <NA>          NA          NA          NA      NA
## 3          <NA>          NA          NA          NA      NA
## 4 ENSMUSG00000000056.bs1          3          3      TRUE  TRUE
## 5 ENSMUSG00000000078.bs1          3          3      TRUE FALSE
## 6          <NA>          NA          NA          NA      NA
##  rownumber  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
## 1      1261  0  0  0  0  1  1  1  1  0  0  1  1  1  1  1  0  1  1  1  1  0  0
## 2      NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
## 3      12209  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1  1  0
## 4      4440  0  0  0  0  0  0  0  0  0  0  1  0  1  1  1  1  1  1  1  1  1  1
## 5      4954  1  1  1  1  1  1  1  1  1  1  0  1  1  1  1  1  1  0  0  1  1  1
## 6      NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
## 23 heat_ksA$cluster
## 1  0          3
## 2 NA      Non-target
## 3  0          5
## 4  1          5
## 5  1          1
## 6 NA      Non-target
```

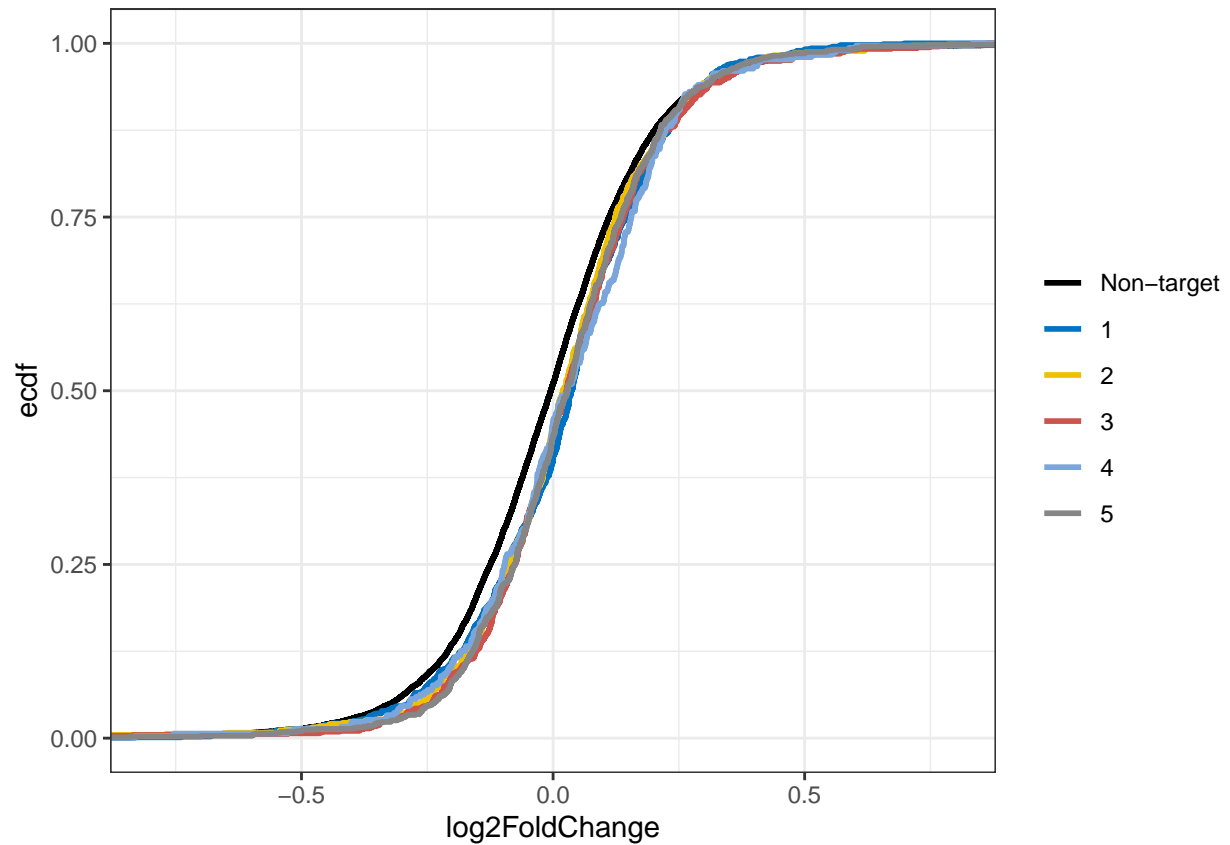
```
table(RPF$`heat_ksA$cluster`)
```

```
##
##          1          2          3          4          5 Non-target
##      837      629      858      296      885      7864
```

plot ecdf

```
#RNA
RNAumplot = ggplot(RNA, aes(log2FoldChange, colour=factor(`heat_ksA$cluster`,
                                                    levels = c("Non-target", "1", "2", "3"
stat_ecdf(geom = "step", linewidth=1) +
coord_cartesian(xlim = c(-0.8, 0.8)) +
scale_colour_manual(values = c("black", farbe1, farbe2, farbe3, farbe4, farbe5)) +
theme_bw() +
theme(legend.title = element_blank())

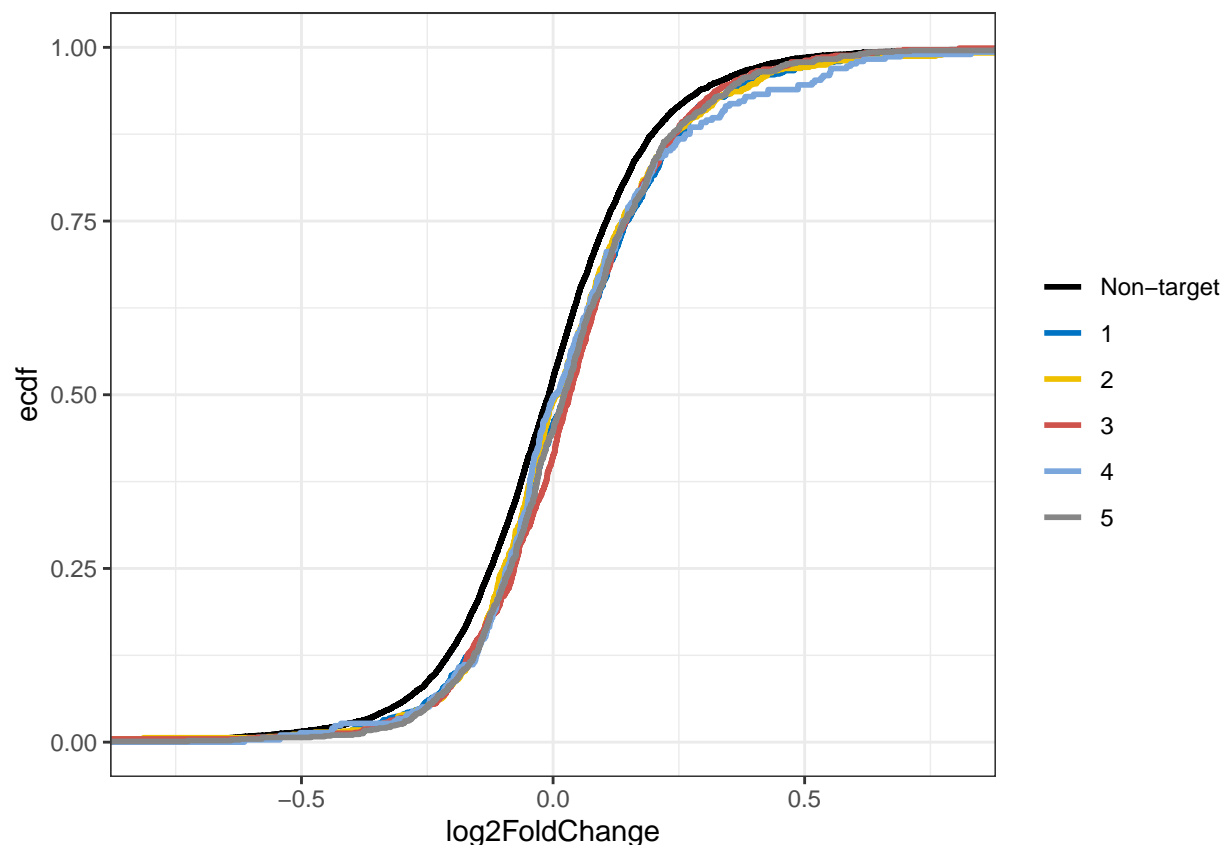
RNAumplot
```



```
#RPF
```

```
RPFnumplot = ggplot(RPF, aes(log2FoldChange, colour=factor(`heat_ksA$cluster`,
                                                             levels = c("Non-target", "1", "2", "3"
stat_ecdf(geom = "step", linewidth=1) +
coord_cartesian(xlim = c(-0.8, 0.8)) +
scale_colour_manual(values = c("black", farbe1, farbe2, farbe3, farbe4, farbe5)) +
theme_bw() +
theme(legend.title = element_blank())
```

```
RPFnumplot
```



session info

```
sessionInfo()
```

```
## R version 4.2.3 (2023-03-15 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=German_Germany.utf8  LC_CTYPE=German_Germany.utf8
## [3] LC_MONETARY=German_Germany.utf8 LC_NUMERIC=C
## [5] LC_TIME=German_Germany.utf8
##
## attached base packages:
## [1] grid      stats4    stats      graphics  grDevices  utils      datasets
## [8] methods  base
##
## other attached packages:
## [1] ComplexHeatmap_2.15.2      circlize_0.4.15
## [3] seqinr_4.2-30              ggplot2_3.4.2
## [5] dplyr_1.1.1                BSgenome.Mmusculus.UCSC.mm10_1.4.3
## [7] BSgenome_1.66.3            rtracklayer_1.58.0
## [9] Biostrings_2.66.0          XVector_0.38.0
```



```

## [11] GenomicRanges_1.50.2          GenomeInfoDb_1.34.9
## [13] IRanges_2.32.0                 S4Vectors_0.36.2
## [15] BiocGenerics_0.44.0
##
## loaded via a namespace (and not attached):
## [1] MatrixGenerics_1.10.0          Biobase_2.58.0
## [3] foreach_1.5.2                  highr_0.10
## [5] GenomeInfoDbData_1.2.9        Rsamtools_2.14.0
## [7] yaml_2.3.7                     pillar_1.9.0
## [9] lattice_0.20-45               glue_1.6.2
## [11] digest_0.6.31                 RColorBrewer_1.1-3
## [13] colorspace_2.1-0              htmltools_0.5.4
## [15] Matrix_1.5-3                  XML_3.99-0.14
## [17] pkgconfig_2.0.3               GetoptLong_1.0.5
## [19] magick_2.7.4                  zlibbioc_1.44.0
## [21] scales_1.2.1                  BiocParallel_1.32.6
## [23] tibble_3.2.1                  farver_2.1.1
## [25] generics_0.1.3                withr_2.5.0
## [27] SummarizedExperiment_1.28.0   cli_3.6.0
## [29] magrittr_2.0.3                crayon_1.5.2
## [31] evaluate_0.20                 fansi_1.0.4
## [33] doParallel_1.0.17            MASS_7.3-58.2
## [35] Cairo_1.6-0                   tools_4.2.3
## [37] GlobalOptions_0.1.2          BiocIO_1.8.0
## [39] lifecycle_1.0.3              matrixStats_0.63.0
## [41] mgsub_1.7.3                   munsell_0.5.0
## [43] cluster_2.1.4                 DelayedArray_0.23.2
## [45] ade4_1.7-22                   compiler_4.2.3
## [47] rlang_1.1.0                   RCurl_1.98-1.12
## [49] iterators_1.0.14             rstudioapi_0.14
## [51] rjson_0.2.21                  labeling_0.4.2
## [53] bitops_1.0-7                  rmarkdown_2.21
## [55] restfulr_0.0.15               gtable_0.3.3
## [57] codetools_0.2-19             R6_2.5.1
## [59] GenomicAlignments_1.34.1     knitr_1.42
## [61] fastmap_1.1.1                 utf8_1.2.3
## [63] clue_0.3-64                   shape_1.4.6
## [65] parallel_4.2.3                Rcpp_1.0.10
## [67] vctrs_0.6.1                   png_0.1-8
## [69] tidyselect_1.2.0             xfun_0.37

```