Peak annotation and filtering

Riccardo Mosca

2025-05-12

This markdown filters and annotates the RAPseq peak bed file obtained from Snakemake pipeline, according to Halo and Input signal, computes the peak binding score (BS) and annotation. The output peak file is saved as .txt.

Loading packages and annotation file

```
library(dplyr)
library(stringr)
library(GenomicRanges)
library(GenomicFeatures)
library(idr)

#
txdb <- makeTxDbFromGFF(file =
"/Users/riccardomosca/Desktop/RAPseq_PAPER/ANNOTATIONs/gencode.v37.annotation
.gtf",
    format = "gtf")
Gencode_v37_IDs <- read.table(file =
"/Users/riccardomosca/Desktop/RAPseq_PAPER/ANNOTATIONs/gencode.v37.IDs.txt")
colnames(Gencode_v37_IDs) <- c("gene_ID", "transcript_ID", "gene_strand",
"gene_name",
    "gene_type")
Gencode_v37_IDs <- Gencode_v37_IDs[!duplicated(Gencode_v37_IDs), ]
Gencode_v37_IDs <- Gencode_v37_IDs[!is.na(Gencode_v37_IDs$transcript_ID), ]
head(Gencode_v37_IDs)
```

```
##             gene_ID     transcript_ID gene_strand   gene_name
## 1 ENSG00000223972.5 ENST00000456328.2           +     DDX11L1
## 2 ENSG00000223972.5 ENST00000450305.2           +     DDX11L1
## 3 ENSG00000227232.5 ENST00000488147.1           -      WASH7P
## 4 ENSG00000278267.1 ENST00000619216.1           -   MIR6859-1
## 5 ENSG00000243485.5 ENST00000473358.1           + MIR1302-2HG
## 6 ENSG00000243485.5 ENST00000469289.1           + MIR1302-2HG
##                            gene_type
## 1 transcribed_unprocessed_pseudogene
## 2 transcribed_unprocessed_pseudogene
## 3             unprocessed_pseudogene
## 4                             miRNA
## 5                           lncRNA
## 6                           lncRNA
```

Building unified annotation of transcript features

```r
# 1. Extract transcript-level features from the TxDb object
Intron_GR <- intronsByTranscript(txdb, use.names = TRUE)
Exon_GR <- exonsBy(txdb, by = "tx", use.names = TRUE)
ThreeUTR_GR <- threeUTRsByTranscript(txdb, use.names = TRUE)
FiveUTR_GR <- fiveUTRsByTranscript(txdb, use.names = TRUE)
CDS_GR <- cdsBy(txdb, by = "tx", use.names = TRUE)
pass_1 <- subsetByOverlaps(Exon_GR, CDS_GR, invert = T)
pass_2 <- subsetByOverlaps(pass_1, ThreeUTR_GR, invert = T)
Exon_GR <- subsetByOverlaps(pass_2, FiveUTR_GR, invert = T)
rm(pass_1)
rm(pass_2)

# 2. Filter exonic ranges to remove CDS, 3'UTR, and 5'UTR regions
pass_1 <- subsetByOverlaps(Exon_GR, CDS_GR, invert = TRUE)
pass_2 <- subsetByOverlaps(pass_1, ThreeUTR_GR, invert = TRUE)
Exon_GR <- subsetByOverlaps(pass_2, FiveUTR_GR, invert = TRUE)
rm(pass_1, pass_2)

# 3. Convert each GRangesList to a data.frame and tag with feature type
make_df <- function(gr, name) {
    df <- as.data.frame(gr)[, c(3, 4, 5, 2, 7)]  # select
(seqnames,start,end,group,strand)
    df$feature <- name  # annotate feature type
    names(df) <- c("chr", "start", "end", "transcript_ID", "feature_strand",
"feature")
    df
}
Introns <- make_df(Intron_GR, "intron")
Exons <- make_df(Exon_GR, "exon")
CDSs <- make_df(CDS_GR, "CDS")
FiveUTRs <- make_df(FiveUTR_GR, "5UTR")
ThreeUTRs <- make_df(ThreeUTR_GR, "3UTR")

# 4. Combine all feature RBP, remove intermediate objects
Features <- rbind(Introns, Exons, CDSs, FiveUTRs, ThreeUTRs)

# 5. Merge with Gencode transcript→gene mapping, reorder and rename columns
Features <- merge(Features, Gencode_v37_IDs, by = "transcript_ID")
Features$gene_ID <- as.character(Features$gene_ID)
Features <- Features[, c(2, 3, 4, 7, 6, 5, 9, 10)]
colnames(Features) <- c("chr", "start", "end", "gene_ID", "feature",
"strand", "gene_name",
    "gene_type")

# 6. Deduplicate and finalize metadata columns
Features$IDs <- with(Features, paste(chr, start, end, gene_ID, feature,
strand, gene_name,
    sep = "_"))
Features <- Features[!duplicated(Features$IDs), 1:8]
```

```
Features$chr <- as.character(Features$chr)
Features$strand <- as.character(Features$strand)

# 7. Re-create GRanges object for the final feature set
Features_GR <- makeGRangesFromDataFrame(Features[, 1:6])

# 8. Tidy up column names and types
colnames(Features) <- c("chr", "start", "end", "gene_ID", "feature",
"gene_strand",
    "gene_name", "gene_type")
Features$gene_type <- as.character(Features$gene_type)
```

Filtering and annotation steps

```
# Get the list of peak files in the directory
peak_files <-
list.files("/Users/riccardomosca/Desktop/RAPseq_PAPER/PEAKs/T7_Fig5/",
full.names = TRUE)

# Create an empty list to store results for each RBP
results_list <- list()

# Loop through each peak file
for (file in peak_files) {
  # Extract RBP name from file path
  protein_name <- tools::file_path_sans_ext(basename(file))

  # Read bed file
  RBP <- read.table(file, sep = '\t', header = TRUE)


  colnames(RBP) <- c("chr","Summit_start","Summit_end", "start", "end",
"strand","Count_rep1","Count_rep2","minuslog10pval_rep1","minuslog10pval_rep2
","Rep1","Rep2","Halo","Input","positive_fa","negative_fa")




####### the few variables created bellow are required by the idr package to
compute the irreproducibility discovery rate #######
mu <- 2.6
sigma <- 1.3
rho <- 0.8
p <- 0.7
```

```
############. If the user wants, the few lines bellow can be used to compute
IDRs (both local and global) before any filtering is done #######
x<-RBP[,c("minuslog10pval_rep1","minuslog10pval_rep2")]
idr.out <- est.IDR(x, mu, sigma, rho, p, eps=0.001, max.ite=30)
RBP$local_idr <- idr.out$idr
RBP$global_IDR <- idr.out$IDR

RBP$RBP <- tools::file_path_sans_ext(basename(file))
RBP$Peak_ID <- paste( RBP$chr,RBP$start,RBP$end,RBP$strand, sep = "_" )
RBP$Summit_start <-RBP$Summit_start
RBP$Summit_end <-RBP$Summit_end
RBP$IDs <- paste( RBP$chr,RBP$Summit_start,RBP$Summit_end,RBP$strand, sep =
"_" )
RBP$Halo[RBP$Halo == 0] <- min(RBP[RBP$Halo!=0,"Halo"])
RBP$Input[RBP$Input == 0] <- min(RBP[RBP$Input!=0,"Input"])
RBP$minuslog10FDR_rep1 <- round(-log10(p.adjust(10^-RBP$minuslog10pval_rep1,
method = "BH")),5)
RBP$minuslog10FDR_rep2 <- round(-log10(p.adjust(10^-RBP$minuslog10pval_rep2,
method = "BH")),5)

##### Filtering: FDR <= 0.05; Fold Change: above Halo > 1 and above Input >
1; Fold Change Halo over Input < 2  ##############
RBP <- RBP[RBP$minuslog10FDR_rep1 >= 1.30103,] #14097
RBP <- RBP[RBP$minuslog10FDR_rep2 >= 1.30103,] #13950

RBP <- RBP[RBP$minuslog10pval_rep1 >= 4,] #11976
RBP <- RBP[RBP$minuslog10pval_rep2 >= 4,] #11072

RBP <-RBP[RBP$Halo/RBP$Input < 2,] #9053



RBP$FCH_rep1 <-RBP$Rep1/RBP$Halo
RBP$FCH_rep2 <-RBP$Rep2/RBP$Halo
RBP$FCI_rep1 <-RBP$Rep1/RBP$Input
RBP$FCI_rep2 <-RBP$Rep2/RBP$Input
RBP$FCmean_rep1 <- (RBP$FCH_rep1 +RBP$FCI_rep1)/2
RBP$FCmean_rep2 <- (RBP$FCH_rep2 +RBP$FCI_rep2)/2
RBP$BS_rep1 <- log2(RBP$Rep1) *RBP$FCmean_rep1
RBP$BS_rep2 <- log2(RBP$Rep2) *RBP$FCmean_rep2
RBP$BS <- (RBP$BS_rep1 +RBP$BS_rep2)/2
RBP$Mean_FCH <- (RBP$FCH_rep1 +RBP$FCH_rep2)/2
RBP$Mean_FCI <- (RBP$FCI_rep1 +RBP$FCI_rep2)/2

RBP <-RBP[RBP$FCH_rep1 > 1 &RBP$FCH_rep2 > 1,] #9051
RBP <-RBP[RBP$FCI_rep1 > 1 &RBP$FCI_rep2 > 1,] #9046
```

```
##### Filtering for sequencing complexity to account for spurious and
artifactual alignments, GA dinucleotide used for complexity determination
#####
RBP$positive_fa_check <- str_sub(RBP$positive_fa,85,115)
Gs <- str_count(RBP$positive_fa_check,"G") / (115-85)
As <- str_count(RBP$positive_fa_check,"A") / (115-85)
Ts <- str_count(RBP$positive_fa_check,"T") / (115-85)
Cs <- 1 - Gs - As - Ts
GAs <- Gs + As
RBP$GAs <- GAs
RBP$GTs <- Gs + Ts
RBP$GCs <- Gs + Cs
RBP$GATs <- Gs + As + Ts
GAs <- Gs + As
GTs <- Gs + Ts
GATs <- Gs + As + Ts
RBP <- RBP[RBP$GAs <= 0.7,]
RBP <- RBP[RBP$GATs <= 0.9,]

    ####### Peak Annotation ########
   GR <-
makeGRangesFromDataFrame(RBP[,c("chr","Summit_start","Summit_end","Peak_ID","
RBP","strand")])

   RBP <-RBP[as.data.frame(findOverlaps(GR,Features_GR, type =
"within"))[,1],]
   Annots <- Features[as.data.frame(findOverlaps(GR,Features_GR, type =
"within"))[,2],][,4:8]
   RBP <- cbind(RBP,Annots)
   RBP$strand <- as.character(RBP$strand)
   RBP$gene_strand <- as.character(RBP$gene_strand)
   RBP <-RBP[RBP$strand ==RBP$gene_strand, ]
   RBP$IDs <- paste(RBP$IDs,RBP$feature,RBP$gene_ID, sep = "_" )
   RBP <-RBP[duplicated(RBP$IDs) == "FALSE",]
   RBP$IDs <-
paste(RBP$chr,RBP$Summit_start,RBP$Summit_end,RBP$strand,RBP$feature, sep =
"_" )
   RBP <-RBP[duplicated(RBP$IDs) == "FALSE",]

   RBP$IDs <- paste(RBP$chr,RBP$Summit_start,RBP$Summit_end,RBP$strand, sep =
"_" )
   RBP$Unique_Anno <-  ave( seq_along(RBP$IDs),RBP$IDs, FUN = length ) == 1
   RBP <-RBP[(RBP$Unique_Anno == "FALSE" &RBP$feature == "intron") == "FALSE"
,]
    #print(paste(i,nrow(RBP)))

   RBP$Unique_Anno <-  ave( seq_along(RBP$IDs),RBP$IDs, FUN = length ) == 1
   RBP <-RBP[(RBP$Unique_Anno == "FALSE" &RBP$feature == "exon") == "FALSE"
,]
```

```
    #print(paste(i,nrow(RBP)))

   RBP$Unique_Anno <-  ave( seq_along(RBP$IDs),RBP$IDs, FUN = length ) == 1
   RBP <-RBP[(RBP$Unique_Anno == "FALSE" &RBP$feature == "5UTR") == "FALSE"
,]
   # print(paste(i,nrow(RBP)))

   RBP$Unique_Anno <-  ave( seq_along(RBP$IDs),RBP$IDs, FUN = length ) == 1
   RBP <-RBP[(RBP$Unique_Anno == "FALSE" &RBP$feature == "3UTR") == "FALSE"
,]
   # print(paste(i,nrow(RBP)))

   RBP$peak_uniqueness <- ave( seq_along(RBP$Peak_ID),RBP$Peak_ID, FUN =
length ) == 1
    #print(paste(i,nrow(RBP)))

    AAA <-RBP[,c("gene_ID","BS")]
    BBB <- AAA %>% group_by(gene_ID) %>% summarise(Gene_BI = sum(BS))
    BBB <- as.data.frame(BBB)

   RBP <-RBP[grep("pseudogene",RBP$gene_type,invert = T),]
   RBP <-RBP[grep("tRNA|rRNA",RBP$gene_type,invert = T),]
   #print(paste(i,nrow(RBP)))

   RBP <- merge(RBP,BBB,by = "gene_ID")









  round(cor(RBP$FCH_rep1,RBP$FCH_rep2,method = "spearman"),2)




Columns <- c("chr", "start", "end", "Peak_ID", "RBP", "strand",
"Summit_start",
"Summit_end","Rep1","Rep2","Halo","Input","minuslog10pval_rep1",
"minuslog10pval_rep2", "minuslog10FDR_rep1", "minuslog10FDR_rep2",
"FCH_rep1", "FCH_rep2", "FCI_rep1", "FCI_rep2", "Mean_FCH", "Mean_FCI", "BS",
"gene_ID", "gene_name", "gene_type", "feature", "Gene_BI", "local_idr",
```

```
"global_IDR", "positive_fa", "negative_fa" )

Columns_rename <- c("chr", "start", "end", "Peak_ID", "RBP", "strand",
"Summit_start",
"Summit_end","Rep1","Rep2","Halo","Input","minuslog10pval_rep1",
"minuslog10pval_rep2", "minuslog10FDR_rep1", "minuslog10FDR_rep2",
"FCH_rep1", "FCH_rep2", "FCI_rep1", "FCI_rep2", "Mean_FCH", "Mean_FCI", "BS",
"gene_ID", "gene_name", "gene_type", "feature", "Gene_BS", "local_idr",
"global_IDR", "positive_fa", "negative_fa" )



  RBP <-RBP[,Columns]
  colnames(RBP) <- Columns_rename



  results_list[[protein_name]] <- RBP
  write.table(RBP, file =
paste0("/Users/riccardomosca/Desktop/RAPseq_PAPER/PEAKs/ANNOTATED/T7_Fig5/NEW
/", protein_name, ".txt"), row.names = FALSE, col.names = TRUE, sep = "\t")
}

head(RBP)

##     chr     start       end                     Peak_ID        RBP strand
## 1  chrX 100635685 100635723 chrX_100635685_100635723_- Ybx1_final      -
## 2  chr7  92117179  92117212   chr7_92117179_92117212_- Ybx1_final      -
## 3  chr4  17597076  17597110   chr4_17597076_17597110_+ Ybx1_final      +
## 4 chr17  38919207  38919282  chr17_38919207_38919282_+ Ybx1_final      +
## 5 chr17  38919356  38919395  chr17_38919356_38919395_+ Ybx1_final      +
## 6  chr8  17550315  17550353   chr8_17550315_17550353_+ Ybx1_final      +
##   Summit_start Summit_end    Rep1     Rep2 Halo     Input
minuslog10pval_rep1
## 1    100635706  100635714 12.4950 21.46333  3.0  4.270000
7.99793
## 2     92117200   92117208 46.4100 27.67400 21.0 18.300000
21.32407
## 3     17597084   17597092 17.6120 18.49500 11.4  9.760000
8.05460
## 4     38919228   38919236 34.0340 40.41500 20.5 19.215000
9.99117
## 5     38919360   38919368 22.0150 23.97500 14.0 12.810000
7.99793
## 6     17550327   17550335 16.9575 18.49500  9.0  9.353333
6.95302
##   minuslog10pval_rep2 minuslog10FDR_rep1 minuslog10FDR_rep2 FCH_rep1
FCH_rep2
## 1            18.79928            7.43562           18.07908 4.165000
```

```
7.154444
## 2             4.14464           19.78318            4.04329 2.210000
1.317810
## 3             7.99793            7.45417            7.72918 1.544912
1.622368
## 4            21.10764            9.24691           20.30951 1.660195
1.971463
## 5            11.34488            7.43562           10.92380 1.572500
1.712500
## 6             9.08008            6.48219            8.76457 1.884167
2.055000
##   FCI_rep1 FCI_rep2 Mean_FCH Mean_FCI        BS             gene_ID
gene_name
## 1 2.926230 5.026542 5.659722 3.976386 19.930400 ENSG00000000003.15
TSPAN6
## 2 2.536066 1.512240 1.763905 2.024153  9.958296 ENSG00000001630.17
CYP51A1
## 3 1.804508 1.894980 1.583640 1.849744  7.166568 ENSG00000002549.13
LAP3
## 4 1.771220 2.103305 1.815829 1.937263  9.802112 ENSG00000002834.18
LASP1
## 5 1.718579 1.871585 1.642500 1.795082  7.776771 ENSG00000002834.18
LASP1
## 6 1.812990 1.977370 1.969583 1.895180  8.017785 ENSG00000003989.18
SLC7A2
##        gene_type feature   Gene_BS    local_idr   global_IDR
## 1 protein_coding     CDS 19.930400 1.662609e-05 2.618438e-06
## 2 protein_coding     CDS  9.958296 1.248706e-05 2.025971e-06
## 3 protein_coding     CDS  7.166568 1.142055e-03 1.602190e-04
## 4 protein_coding    3UTR 17.578883 2.686873e-06 4.370830e-07
## 5 protein_coding    3UTR 17.578883 2.253625e-04 3.493414e-05
## 6 protein_coding     CDS  8.017785 1.549546e-03 2.193270e-04
##
positive_fa
## 1
TGTGATTTGAAGATGCTGCTGTACACAGTGCGTAACTGTTTGTATTCTCTGAATTCCCACATAGATCACTGGCGTTA
TCCTTCTTGCAGTTGGCATTTGGGGCAAGGTGAGCCTGGAGAATTACTTTTCTCTTTTAAATGAGAAGGCCACCAAT
GTCCCCTTCGTGCTCATTGCTACTGGTACCGTCATTATTCTTTTGG
## 2
CCATGTATATTCATGCTTTATTACACAATGATCTGATTATTTGTCAGCACATTTTTAAATTCTCTAATGAAATGTGT
GGCTTTTGTTTATAGACTGTGGCAGGGTATACCATTCCTCCAGGACATCAGGTGTGTGTTTCTCCCACTGTCAATCA
AAGACTTAAAGACTCATGGGTAGAACGCCTGGACTTTAATCCTGAT
## 3
AGGTGGCATGTTTGGACCCAGTATATACTGTGTGCCTTCATATATTATTTCCAATAGGTCTGGCCCCTCTTTGTGAA
AATATGCCCAGCGGCAAGGCCAACAAGCCGGGGGATGTTGTTAGAGCCAAAAACGGGAAGACCATCCAGGTTTGTAA
ATGTGAGACACAGCACTCCCCATCCAGCGTTCCTCAGGAATCCCGT
## 4
ATTCCAGGGCTGGGGTGAGCCTGACTGCCAGGACCCCAGGTCAGGGGCTCCCTACATTCCCCAGAGTGGGATCCACT
TCTTGGTTCCTGGGATGGCGATGGGGACTCTGCCGCTGTGTAGGGACCAGTGGGATGGGCTCTACCTCTCTTTCTCA
AAGAGGGGGCTCTGCCCACCTGGGGTCTCTCTCCCTACCTCCCTCC
```

```
## 5
TGGGCTCTACCTCTCTTTCTCAAAGAGGGGGCTCTGCCCACCTGGGGTCTCTCTCCCTACCTCCCTCCTCAGGGGCA
ACAACAGGAGAATGGGGTTCCTGCTGTGGGGCGAATTCATCCCCTCCCCGCGCGTTCCTTCGCACACTGTGATTTTG
CCCTCCTGCCCACGCAGACCTGCAGCGGGCAAAGAGCTCCCGAGGA
## 6
TGAGAAAAGTCACCAGAAGGAGAGTTTTCTGTCAAAGTGACTTTCCAATGTGTTTGTTCTCTTTCTTAGAGAGCCAC
CTTCTGAAAACGGAACAAGTATCTATGGGGCTGGTGGCTTTATGCCTTATGGCTTTACGGGAACGTTGGCTGGTGCT
GCAACTTGCTTTTATGCCTTTGTGGGATTTGACTGCATTGCAACAA
##
negative_fa
## 1
TCTCAGTTGTGGACGCTCGTAAGTTTTCGGCAGTTTCCGGGGAGACTCGGGGACTCCGCGTCTCGCTCTCTGTGTTC
CAATCGCCCGGTGCGGTGGTGCAGGGTCTCGGGCTAGTCATGGCGTCCCCGTCTCGGAGACTGCAGACTAAACCAGT
CATTACTTGTTTCAAGAGCGTTCTGCTAATCTACACTTTTATTTTC
## 2
GCACATGCCTGTGGTCCCAGCTGCTTGGGAGGCTGATGTGGGTAGATTGCTTGAGCCTGGGAGGTTGAAACTGCAGT
AATCCATGATCACGCCACTGCATTGCTGCCTGGGCAACAGAGCAAGACCCTGTCTCAAAAAAAAATAAAAATAAAAAA
GAAAGAAAAAGAAAAGAGGAAGTAGAGTAGCATAAAAGAGATTTTT
## 3
ATGTAGCTGTAGCTTTCTTAAAGCTCTTTAACCTCATTTTTGGAAATCTTTACATTTTTCCCCTTCTTGTTACAAAG
TTACGGGTAAAATGGACTAGACATTTTTCTATTTATTTTGGCTTCCAAAGTCATCAACATTAGGTCATCAGCTCTGG
TACAGTGATTATTATATTATTATTTTTTTATTTTTAATTAGAAATG
## 4
GGTGTGGAGTTGGGGCTGCCATAGGGTCTGCAGCCTGCTGGGGCTAAGCGGTGGAGGAAGGCTCTGTCACTCCAGGC
ATATGTTTCCCCATCTCTGTCTGGGGCTACAGAATAGGGTGGCAGAAGTGTCACCCTGTGGGTGTCTCCCTCGGGGG
CTCTTCCCCTAGACCTCCCCCTCACTTACATAAAGCTCCCTTGAAG
## 5
CTGTGGGTGTCTCCCTCGGGGGCTCTTCCCCTAGACCTCCCCCTCACTTACATAAAGCTCCCTTGAAGCAAGAAAGA
GGGTCCCAGGGCTGCAAAACTGGAAGCACAGCCTCGGGGATGGGGAGGGAAAGACGGTGCTATATCCAGTTCCTGCT
CTCTGCTCATGGGTGGCTGTGACAACCCTGGCCTCACTTGATTCAT
## 6
CCTCTTCTCCCTCTGGGAATCGGATTTTTACGTAATAAGAAAGTACCTTCAAGAATAGAGGAAATTTCATGTAAAAC
AGATGGATCAGCTGGGCACAGTGGCTCACGCCTGTAATCCCAATCTTTGGGACATTGAGGAGGGTGGATCACTTGAG
GCCAGGAATTAGAGACTAGACTGGCCAACATGGTGCAAACCATCTC
```

```
sessionInfo()
```

```
## R version 4.3.2 (2023-10-31)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Sonoma 14.5
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.3-
arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-
arm64/Resources/lib/libRlapack.dylib;  LAPACK version 3.11.0
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
```

```
## time zone: Europe/Stockholm
## tzcode source: internal
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] idr_1.3                GenomicFeatures_1.54.4 AnnotationDbi_1.64.1
##  [4] Biobase_2.62.0         GenomicRanges_1.54.1   GenomeInfoDb_1.38.8
##  [7] IRanges_2.36.0         S4Vectors_0.40.2       BiocGenerics_0.48.1
## [10] stringr_1.5.1          dplyr_1.1.4
##
## loaded via a namespace (and not attached):
##  [1] SummarizedExperiment_1.32.0 KEGGREST_1.42.0
##  [3] rjson_0.2.23                xfun_0.52
##  [5] lattice_0.22-7              vctrs_0.6.5
##  [7] tools_4.3.2                 bitops_1.0-9
##  [9] generics_0.1.3              parallel_4.3.2
## [11] curl_6.2.2                  tibble_3.2.1
## [13] RSQLite_2.3.11              blob_1.2.4
## [15] pkgconfig_2.0.3             Matrix_1.6-5
## [17] dbplyr_2.5.0                lifecycle_1.0.4
## [19] GenomeInfoDbData_1.2.11     compiler_4.3.2
## [21] Rsamtools_2.18.0            Biostrings_2.70.3
## [23] progress_1.2.3              codetools_0.2-20
## [25] htmltools_0.5.8.1           RCurl_1.98-1.17
## [27] yaml_2.3.10                 pillar_1.10.2
## [29] crayon_1.5.3                BiocParallel_1.36.0
## [31] DelayedArray_0.28.0         cachem_1.1.0
## [33] abind_1.4-8                 tidyselect_1.2.1
## [35] digest_0.6.37               stringi_1.8.7
## [37] restfulr_0.0.15             grid_4.3.2
## [39] biomaRt_2.58.2              fastmap_1.2.0
## [41] SparseArray_1.2.4           cli_3.6.5
## [43] magrittr_2.0.3              S4Arrays_1.2.1
## [45] XML_3.99-0.18               prettyunits_1.2.0
## [47] filelock_1.0.3              rappdirs_0.3.3
## [49] bit64_4.6.0-1               rmarkdown_2.29
## [51] XVector_0.42.0              httr_1.4.7
## [53] matrixStats_1.5.0           bit_4.6.0
## [55] png_0.1-8                   hms_1.1.3
## [57] memoise_2.0.1               evaluate_1.0.3
## [59] knitr_1.50                  BiocIO_1.12.0
## [61] BiocFileCache_2.10.2        rtracklayer_1.62.0
## [63] rlang_1.1.6                 glue_1.8.0
## [65] DBI_1.2.3                   formatR_1.14
## [67] xml2_1.3.8                  rstudioapi_0.17.1
## [69] R6_2.6.1                    MatrixGenerics_1.14.0
## [71] GenomicAlignments_1.38.2    zlibbioc_1.48.2
```