GUIDEseq user's guide

Lihua Julie Zhu [1], ***Michael Lawrence*** [1], ***Ankit Gupta*** [1], ***Herve Pages*** [1], ***Alper Kucukural*** [1], ***Manuel Garber*** [1], ***and Scot Wolfe*** [1]

[1]University of Massachusetts Chan Medical School

26 May 2022

Abstract

Instructions on using GUIDEseq in analyzing GUIDEseq data.

Package

GUIDEseq 1.27.3

Contents

1 Introduction

The most recently developed genome editing system, CRISPR-Cas9 has greater inherent flexibility than prior programmable nuclease platforms because sequence-specific recognition resides primarily within the associated sgRNA, which permits a simple alteration of its recognition sequence. The short Protospacer Adjacent Motif (PAM), which is recognized by Cas9, is the chief constraint on the target site design density. Because of its simplicity and efficacy, this technology is revolutionizing biological studies and holds tremendous promise for therapeutic applications (Ledford 2015; Cox, Platt, and Zhang 2015).

However, imperfect cleavage specificity of CRISPR/Cas9 nuclease within the genome is a cause for concern for its therapeutic application. S. pyogenes Cas9 (SpyCas9)-based nucleases can cleave an imperfect heteroduplex formed between the guide sequence and a DNA sequence containing a functional PAM where the number, position and type of base mismatches can impact the level of activity (Mali et al. 2013; Hsu et al. 2013; Fu et al. 2013). This degree of promiscuity is problematic for therapeutic applications, since the generation of DNA breaks at unintended (off-target) sites has the potential to alter gene expression and function through direct mutagenesis or the generation of genomic rearrangements. Experimentally defining the number and activity of off-target sites for a given Cas9/sgRNA complex genome-wide is critical to assess and improve nuclease precision.

A new suite of genome-wide off-target detection methods have recently been described that can identify sites with low cleavage activity within a population of nuclease-treated cells. One of the most sensitive and straightforward methods to employ is GUIDE-seq (Tsai et al. 2015). This method relies on erroneous NHEJ-mediated DNA repair to capture co-introduced blunt-ended double stranded oligonucleotides (dsODNs) at Cas9-induced breakpoints within the genome. The GUIDE-seq dsODNs display high insertion frequency (up to 50% of the measured indel rate (Tsai et al. 2015)) at Cas9-induced DSBs, thereby tagging these loci for selective amplification and subsequent deep sequencing. The method is quite sensitive as off-target sites with $>0.1\%$ indel frequency can be detected, and the frequency of dsODN insertion appears to be correlated with the frequency of Cas9-induced lesions at each site (Tsai et al. 2015). This method has been used successfully to evaluate the precision of Cas9 and its variants (tru-sgRNAs (Tsai et al. 2015) or PAM variants (Kleinstiver et al. 2015)). Given its favorable properties, GUIDE-seq could become a standard in the nuclease field for off-target analysis.

While the GUIDE-seq method is straightforward to employ, to date no bioinformatic tools have been released to the community to support the analysis of this data. We developed *GUIDEseq* package (Zhu et al. 2017) to facilitate the analysis of GUIDE-seq dataset, including retaining one read per unique molecular identifier (UMI), filtering reads lacking integration oligo sequence (dsODNs), identifying peak locations (cleavage sites) and heights, merging cleavage sites from plus strand and those from minus strand, and performing target and off target search of the input gRNA. This analysis leverages our *ChIPPeakAnno* package (Zhu et al. 2010) for merging cleavage sites from plus strand and minus strand, and *CRISPRseek* package (Zhu et al. 2014) for defining the homology of any identified off-target site to the guide sequence and Cas9 PAM specificity.

2 Workflow of GUIDE-seq data analysis

Here is the workflow of GUIDE-seq data analysis with human sequence. First load *GUIDEseq* and *BSgenome.Hsapiens.UCSC.hg19*.

To find BSgenome of other species, please refer to available.genomes in the *BSgenome* package. For example, *BSgenome.Hsapiens.UCSC.hg19* for hg19, *BSgenome.Mmusculus.UCSC.mm10* for mm10, *BSgenome.Celegans.UCSC.ce6* for ce6, *BSgenome.Rnorvegicus.UCSC.rn5* for rn5, *BSgenome.Drerio.UCSC.danRer7* for Zv9, and *BSgenome.Dmelanogaster.UCSC.dm3* for dm3

Then specify the alignment file path as alignment.inputfile, and a umi file path as umi.inputfile containing unique molecularidentifier for each sequence.

```
library(GUIDEseq)
umifile <- system.file("extdata", "UMI-HEK293_site4_chr13.txt", package = "GUIDEseq")
bedfile <- system.file("extdata","bowtie2.HEK293_site4_chr13.sort.bed", package = "GUIDEseq")
bamfile <- system.file("extdata","bowtie2.HEK293_site4_chr13.sort.bam", package = "GUIDEseq")
```

The alignment.inputfile is an alignment file in bed format containing CIGAR information. The alignment.inputfile contains chromosome, start, end, readID, mapping quality, strand and CIGAR information as a tab delimited file. Here is an example line.

```
chr13 27629253 27629403 HWI-M01326:156:1:113:4572:6938/1 44 + 150M
```

Scripts for bin reads, remove adaptor, mapping to genome are available at GUIDE-seq.

The umi.inputfile is a tab delimited file containing at least two columns, read IDs and corresponding unique molecular identifiers (UMI). Script for creating umi.inputfile is available as *getUmi.pl*. An example input file is at testGetUmi. Please make sure to use R1 reads as input to *getUmi.pl*.

2.1 Step 1: Remove PCR bias and obtain unique cleavage events

PCR amplification often leads to biased representation of the starting sequence population. To track the sequence tags present in the initial sequence library,unique molecular identifiers (UMI) are added to the 5 prime of each sequence in the staring library. The function getUniqueCleavageEvents uses the UMI sequence in the umi.inputfile (optionally contains umi plus the first few sequence from R1 reads) to obtain the starting sequence library. It also filters out reads that does not contain the integration oligo sequence, too short or not in the right paired configuration.

For detailed parameter settings for function getUniqueCleavageEvents, please type `help(getUniqueCleavageEvents)` or `?getUniqueCleavageEvents`.

```
uniqueCleavageEvents <- getUniqueCleavageEvents(bamfile, umifile, n.cores.max = 1)
uniqueCleavageEvents$cleavage.gr
## GRanges object with 3841 ranges and 1 metadata column:
##          seqnames    ranges strand |     total
##             <Rle> <IRanges>  <Rle> | <numeric>
##      [1]    chr13  69979989      + |         1
##      [2]    chr13  39262929      + |         1
##      [3]    chr13  94623776      + |         1
##      [4]    chr13  28089785      + |         1
##      [5]    chr13  27629410      + |         1
##      ...      ...       ...    ... .       ...
##   [3837]    chr13  51894561      - |         1
```

```
##   [3838]    chr13  93491789      - |         1
##   [3839]    chr13  27629409      - |         1
##   [3840]    chr13  98262510      - |         1
##   [3841]    chr13  37190177      - |         1
##   -------
##   seqinfo: 25 sequences from an unspecified genome; no seqlengths
```

2.2 Step 2: Summarize cleavage events

Calling the function getPeaks with the results from getUniqueCleavageEvents outputs summarized cleavage events for each moving window with at least `min.reads` of cleavage events.

By default, `window.size` is set to 20, `step` is set to 20, and `min.reads` is set to 2. For detailed parameter settings using function getPeaks, please type `help(getPeaks)`.

```
peaks <- getPeaks(uniqueCleavageEvents$cleavage.gr, min.reads = 80)
peaks.gr <- peaks$peaks
peaks.gr
## GRanges object with 4 ranges and 5 metadata columns:
##       seqnames            ranges strand |     count        bg      p.value
##          <Rle>         <IRanges>  <Rle> | <integer> <numeric>    <numeric>
##   [1]    chr13 27629413-27629420      + |       146     0.584 1.49081e-291
##   [2]    chr13 39262922-39262939      + |       156     0.624 3.20197e-311
##   [3]    chr13 27629400-27629416      - |       103     0.412 5.74504e-207
##   [4]    chr13 39262918-39262920      - |       157     0.636 2.53405e-312
##         SNratio adjusted.p.value
##       <numeric>        <numeric>
##   [1]   250.000     1.49081e-291
##   [2]   250.000     3.20197e-311
##   [3]   250.000     5.74504e-207
##   [4]   246.855     2.53405e-312
##   -------
##   seqinfo: 25 sequences from an unspecified genome; no seqlengths
```

2.3 Step 3: Merge peaks from plus and minus strand

Calling the function mergePlusMinusPeaks with the output from getPeaks to merge peaks from plus strand and minus strand with specific orientation and within certain distance apart.

By default, `plus.strand.start.gt.minus.strand.end` is set to TRUE and `distance`.threshold is set to 40, i.e., twice of the `window.size`. For detailed parameter settings using function mergePlusMinusPeaks, please type `help(mergePlusMinusPeaks)`.

```
mergedPeaks <- mergePlusMinusPeaks(peaks.gr = peaks.gr, output.bedfile = "mergedPeaks.bed")
mergedPeaks$mergedPeaks.gr
## GRanges object with 2 ranges and 2 metadata columns:
##                                                    seqnames            ranges
##                                                       <Rle>         <IRanges>
##   chr13+:27629413:27629420:chr13-:27629400:27629416    chr13 27629400-27629420
##   chr13+:39262922:39262939:chr13-:39262918:39262920    chr13 39262918-39262939
##                                                    strand |     count
```

```
##                                                    <Rle> | <numeric>
##   chr13+:27629413:27629420:chr13-:27629400:27629416     + |       249
##   chr13+:39262922:39262939:chr13-:39262918:39262920     + |       313
##                                                           bg
##                                                    <numeric>
##   chr13+:27629413:27629420:chr13-:27629400:27629416    0.996
##   chr13+:39262922:39262939:chr13-:39262918:39262920    1.260
##   -------
##   seqinfo: 25 sequences from an unspecified genome; no seqlengths

head(mergedPeaks$mergedPeaks.bed)
##   seqnames minStart   maxEnd                                                 names
## 1    chr13 27629400 27629420 chr13+:27629413:27629420:chr13-:27629400:27629416
## 2    chr13 39262918 39262939 chr13+:39262922:39262939:chr13-:39262918:39262920
##   totalCount strand
## 1        249      +
## 2        313      +
```

2.4 Step 4: Off target analysis of extended regions around the identified cleavage sites

Calling the function offTargetAnalysisOfPeakRegions with input gRNA, peaks and genome of interest, to annotate identified cleavage sites with sequence homolgy to input gRNA. For detailed parameter settings using function offTargetAnalysisOfPeakRegions. For more information, please type `help(offTargetAnalysisOfPeakRegions)`.

```
library(BSgenome.Hsapiens.UCSC.hg19)
peaks <- "mergedPeaks.bed" # from step 3
gRNAs <- system.file("extdata","gRNA.fa", package = "GUIDEseq")
outputDir <- getwd()

offTargets <- offTargetAnalysisOfPeakRegions(
                        gRNA = gRNAs,
                        peaks = peaks,
                        format=c("fasta", "bed"),
                        peaks.withHeader = FALSE,
                        BSgenomeName = Hsapiens,
                        outputDir = outputDir,
                        orderOfftargetsBy = "predicted_cleavage_score",
                        allowed.mismatch.PAM = 2,
                        overwrite = TRUE,
                        upstream = 50,
                        downstream = 50,
                        PAM.size = 3,
                        gRNA.size = 20,
                        PAM = "NGG",
                        PAM.pattern = "(NAG|NGG|NGA)$",
                        max.mismatch = 8)
```

```
head(offTargets)
##                   offTarget peak_score
## 1 chr13:-:27629404:27629426        249
## 2 chr13:+:39262912:39262934        313
##                                                names predicted_cleavage_score
## 1 chr13+:27629413:27629420:chr13-:27629400:27629416                        3.2
## 2 chr13+:39262922:39262939:chr13-:39262918:39262920                        1.2
##           name             gRNAPlusPAM      offTarget_sequence
## 1 HEK239_site4 GGCACTGCGGCTGGAGGTGGNGG GGCACTGGGGTTGGAGGTGGGGG
## 2 HEK239_site4 GGCACTGCGGCTGGAGGTGGNGG AGCAGTGCGGCTAGAGGTGGTGG
##   guideAlignment2OffTarget offTargetStrand mismatch.distance2PAM n.mismatch
## 1     .......G..T.........               -                 13,10          2
## 2     A...G.......A.......               +               20,16,8          3
##   targetSeqName offTarget_Start offTarget_End chromosome peak_start peak_end
## 1  HEK239_site4        27629404      27629426      chr13   27629400 27629420
## 2  HEK239_site4        39262912      39262934      chr13   39262918 39262939
##   peak_strand
## 1           +
## 2           +
```

Another example:

```
peaks <- system.file("extdata", "T2plus100OffTargets.bed", package = "CRISPRseek")
gRNAs <- system.file("extdata", "T2.fa", package = "CRISPRseek")
offTargets <- offTargetAnalysisOfPeakRegions(
                        gRNA = gRNAs,
                        peaks = peaks,
                        format=c("fasta", "bed"),
                        peaks.withHeader = TRUE,
                        BSgenomeName = Hsapiens,
                        outputDir = outputDir,
                        orderOfftargetsBy = "predicted_cleavage_score",
                        allowed.mismatch.PAM = 2,
                        overwrite = TRUE,
                        upstream = 50,
                        downstream = 50,
                        PAM.size = 3,
                        gRNA.size = 20,
                        PAM = "NGG",
                        PAM.pattern = "(NAG|NGG|NGA)$",
                        max.mismatch = 8)
```

```
head(offTargets)
##                     offTarget peak_score
## 1 chr10:-:116294250:116294272          5
## 2 chr10:+:135149931:135149953          3
## 3   chr11:+:12308918:12308940         27
## 4   chr11:+:13948354:13948376         22
## 5   chr11:-:31817477:31817499        108
## 6  chr1:+:151031869:151031891         24
##                                            names predicted_cleavage_score
```

```
## 1 chr10+:116294243:116294268chr10-:116294160:116294185                    1.4
## 2 chr10+:135149940:135149965chr10-:135149850:135149875                    0.8
## 3     chr11+:12308937:12308962chr11-:12308837:12308862                    0.1
## 4     chr11+:13948362:13948387chr11-:13948273:13948298                    1.1
## 5     chr11+:31817476:31817501chr11-:31817386:31817411                    5.1
## 6   chr1+:151031877:151031902chr1-:151031782:151031807                    0.1
##   name            gRNAPlusPAM      offTarget_sequence guideAlignment2OffTarget
## 1   T2 GACCCCCTCCACCCCGCCTCNGG CCCCACCCCCACCCCGCCTCAGG     CC..A..C............
## 2   T2 GACCCCCTCCACCCCGCCTCNGG CGCCCTCCCCACCCCGCCTCCGG     CG...T.C............
## 3   T2 GACCCCCTCCACCCCGCCTCNGG ATCCCCCTCCACCCCACCCCTGG     AT............A..C.
## 4   T2 GACCCCCTCCACCCCGCCTCNGG TACCCCCCCCACCCCGCCACAGG     T......C..........A.
## 5   T2 GACCCCCTCCACCCCGCCTCNGG GGGCCCCTCCACCCCGCCTCTGG     .GG................
## 6   T2 GACCCCCTCCACCCCGCCTCNGG CCTCCCCCACACCCCGCATCCGG     CCT....CA........A..
##   offTargetStrand mismatch.distance2PAM n.mismatch targetSeqName
## 1               -           20,19,16,13          4            T2
## 2               +           20,19,15,13          4            T2
## 3               +             20,19,5,2          4            T2
## 4               +               20,13,2          3            T2
## 5               -                 19,18          2            T2
## 6               +      20,19,18,13,12,3          6            T2
##   offTarget_Start offTarget_End chromosome peak_start  peak_end peak_strand
## 1       116294250     116294272      chr10  116294160 116294268           +
## 2       135149931     135149953      chr10  135149850 135149965           +
## 3        12308918      12308940      chr11   12308837  12308962           +
## 4        13948354      13948376      chr11   13948273  13948387           +
## 5        31817477      31817499      chr11   31817386  31817501           +
## 6       151031869     151031891       chr1  151031782 151031902           +
```

Note that in the above example, `peaks.withHeader` is set to TRUE because `T2plus100OffTargets.bed` contains a header line. In contrast, the `mergedPeaks.bed` file created with step3 doesn't contain a header line and `peaks.withHeader` was set to FALSE accordingly.

2.5 Run all steps in one workflow function

The function GUIDEseqAnalysis is a wrapper function that uses the UMI sequence or plus the first few bases of each sequence from R1 reads to estimate the starting sequence library, piles up reads with a user defined window and step size, identify the cleavage sites, merge cleavage sites from plus strand and minus strand, followed by off target analysis of extended regions around the identified cleavage sites. For detailed parameter settings using function GUIDEseqAnalysis, please type `help(GUIDEseqAnalysis)`.

```
gRNA.file <- system.file("extdata", "gRNA.fa", package = "GUIDEseq")
system.time(guideSeqRes <- GUIDEseqAnalysis(
                                                    alignment.inputfile = bamfile,
                                                    umi.inputfile = umifile,
                                                    gRNA.file = gRNA.file,
                                                    orderOfftargetsBy = "peak_score",
                                                    descending = TRUE,
                                                    n.cores.max = 1,
                                                    BSgenomeName = Hsapiens,
                                                    min.reads = 1))
```

```
names(guideSeqRes)
## [1] "offTargets"      "merged.peaks"    "peaks"           "uniqueCleavages"
## [5] "read.summary"
```

3 References

Cox, D. B., R. J. Platt, and F. Zhang. 2015. "Therapeutic Genome Editing: Prospects and Challenges." Journal Article. *Nat Med* 21 (2): 121–31. doi:10.1038/nm.3793.

Fu, Y., J. A. Foden, C. Khayter, M. L. Maeder, D. Reyon, J. K. Joung, and J. D. Sander. 2013. "High-Frequency Off-Target Mutagenesis Induced by CRISPR-Cas Nucleases in Human Cells." Journal Article. *Nat Biotechnol* 31 (9): 822–6. doi:10.1038/nbt.2623.

Hsu, P. D., D. A. Scott, J. A. Weinstein, F. A. Ran, S. Konermann, V. Agarwala, Y. Li, et al. 2013. "DNA Targeting Specificity of RNA-Guided Cas9 Nucleases." Journal Article. *Nat Biotechnol* 31 (9): 827–32. doi:10.1038/nbt.2647.

Kleinstiver, B. P., M. S. Prew, S. Q. Tsai, V. V. Topkar, N. T. Nguyen, Z. Zheng, A. P. Gonzales, et al. 2015. "Engineered CRISPR-Cas9 Nucleases with Altered PAM Specificities." Journal Article. *Nature* 523 (7561): 481–5. doi:10.1038/nature14592.

Ledford, H. 2015. "CRISPR, the Disruptor." Journal Article. *Nature* 522 (7554): 20–24. doi:10.1038/522020a.

Mali, P., J. Aach, P. B. Stranges, K. M. Esvelt, M. Moosburner, S. Kosuri, L. Yang, and G. M. Church. 2013. "CAS9 Transcriptional Activators for Target Specificity Screening and Paired Nickases for Cooperative Genome Engineering." Journal Article. *Nat Biotechnol* 31 (9): 833–8. doi:10.1038/nbt.2675.

Tsai, S. Q., Z. Zheng, N. T. Nguyen, M. Liebers, V. V. Topkar, V. Thapar, N. Wyvekens, et al. 2015. "GUIDE-Seq Enables Genome-Wide Profiling of Off-Target Cleavage by CRISPR-Cas Nucleases." Journal Article. *Nat Biotechnol* 33 (2): 187–97. doi:10.1038/nbt.3117.

Zhu, L. J., C. Gazin, N. D. Lawson, H. Pages, S. M. Lin, D. S. Lapointe, and M. R. Green. 2010. "ChIPpeakAnno: A Bioconductor Package to Annotate ChIP-Seq and ChIP-Chip Data." Journal Article. *BMC Bioinformatics* 11: 237. doi:10.1186/1471-2105-11-237.

Zhu, L. J., B. R. Holmes, N. Aronin, and M. H. Brodsky. 2014. "CRISPRseek: A Bioconductor Package to Identify Target-Specific Guide RNAs for CRISPR-Cas9 Genome-Editing Systems." Journal Article. *PLoS One* 9 (9): e108424. doi:10.1371/journal.pone.0108424.

Zhu, L. J., M. Lawrence, A. Gupta, H. Pages, A. Kucukural, M. Garber, and S. A. Wolfe. 2017. "GUIDEseq: A Bioconductor Package to Analyze GUIDE-Seq Datasets for CRISPR-Cas Nucleases." Journal Article. *BMC Genomics* 18 (1): 379. doi:10.1186/s12864-017-3746-y.

4 Session info

```
sessionInfo()
## R version 4.2.0 (2022-04-22)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.7 LTS
```

```
##
## Matrix products: default
## BLAS:   /usr/lib/libblas/libblas.so.3.6.0
## LAPACK: /usr/lib/lapack/liblapack.so.3.6.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] BSgenome.Hsapiens.UCSC.hg19_1.4.3 BSgenome_1.64.0
##  [3] rtracklayer_1.56.0                Biostrings_2.64.0
##  [5] XVector_0.36.0                    GUIDEseq_1.27.3
##  [7] GenomicRanges_1.48.0              GenomeInfoDb_1.32.1
##  [9] IRanges_2.30.0                    S4Vectors_0.34.0
## [11] BiocGenerics_0.42.0               BiocStyle_2.24.0
##
## loaded via a namespace (and not attached):
##   [1] colorspace_2.0-3            seqinr_4.2-8
##   [3] rjson_0.2.21                ellipsis_0.3.2
##   [5] futile.logger_1.4.3         base64enc_0.1-3
##   [7] ChIPpeakAnno_3.30.0         hash_2.2.6.2
##   [9] bit64_4.0.5                 AnnotationDbi_1.58.0
##  [11] fansi_1.0.3                 xml2_1.3.3
##  [13] splines_4.2.0               cachem_1.0.6
##  [15] knitr_1.39                  zeallot_0.1.0
##  [17] ade4_1.7-19                 jsonlite_1.8.0
##  [19] Rsamtools_2.12.0            dbplyr_2.1.1
##  [21] png_0.1-7                   tfruns_1.5.0
##  [23] graph_1.74.0                BiocManager_1.30.17
##  [25] compiler_4.2.0              httr_1.4.3
##  [27] lazyeval_0.2.2              assertthat_0.2.1
##  [29] Matrix_1.4-1                fastmap_1.1.0
##  [31] limma_3.52.0                cli_3.3.0
##  [33] formatR_1.12                htmltools_0.5.2
##  [35] prettyunits_1.1.1           tools_4.2.0
##  [37] gtable_0.3.0                glue_1.6.2
##  [39] GenomeInfoDbData_1.2.8      dplyr_1.0.9
##  [41] mltools_0.3.5               rappdirs_0.3.3
##  [43] Rcpp_1.0.8.3                Biobase_2.56.0
##  [45] vctrs_0.4.1                 rhdf5filters_1.8.0
##  [47] multtest_2.52.0             xfun_0.30
##  [49] stringr_1.4.0               lifecycle_1.0.1
```

```
##  [51] ensembldb_2.20.1           restfulr_0.0.13
##  [53] XML_3.99-0.9               InteractionSet_1.24.0
##  [55] scales_1.2.0               zlibbioc_1.42.0
##  [57] MASS_7.3-57                ProtGenerics_1.28.0
##  [59] hms_1.1.1                  MatrixGenerics_1.8.0
##  [61] parallel_4.2.0             SummarizedExperiment_1.26.1
##  [63] RBGL_1.72.0                rhdf5_2.40.0
##  [65] AnnotationFilter_1.20.0    lambda.r_1.2.4
##  [67] yaml_2.3.5                 curl_4.3.2
##  [69] memoise_2.0.1              reticulate_1.24
##  [71] ggplot2_3.3.6              keras_2.8.0
##  [73] biomaRt_2.52.0             stringi_1.7.6
##  [75] RSQLite_2.2.13             tensorflow_2.8.0
##  [77] BiocIO_1.6.0               GenomicFeatures_1.48.0
##  [79] filelock_1.0.2             BiocParallel_1.30.0
##  [81] rlang_1.0.2                pkgconfig_2.0.3
##  [83] matrixStats_0.62.0         bitops_1.0-7
##  [85] evaluate_0.15              lattice_0.20-45
##  [87] purrr_0.3.4                Rhdf5lib_1.18.0
##  [89] GenomicAlignments_1.32.0   bit_4.0.4
##  [91] tidyselect_1.1.2           magrittr_2.0.3
##  [93] bookdown_0.26              R6_2.5.1
##  [95] generics_0.1.2             DelayedArray_0.22.0
##  [97] DBI_1.1.2                  pillar_1.7.0
##  [99] whisker_0.4                survival_3.3-1
## [101] KEGGREST_1.36.0            RCurl_1.98-1.6
## [103] tibble_3.1.7               crayon_1.5.1
## [105] futile.options_1.0.1       utf8_1.2.2
## [107] BiocFileCache_2.4.0        rmarkdown_2.14
## [109] progress_1.2.2             grid_4.2.0
## [111] data.table_1.14.2          blob_1.2.3
## [113] digest_0.6.29              VennDiagram_1.7.3
## [115] regioneR_1.28.0            CRISPRseek_1.36.0
## [117] munsell_0.5.0
```