The REDseq user's guide

Lihua Julie Zhu*

April 25, 2023

Contents

1 Introduction

Restriction Enzyme digestion (RED) followed by high throughput sequencing (REDseq) enables genome wide differentiation of highly accessible regions and inaccessible regions. Comparing the profiles of restriction enzyme (RE) digestion among different cell types, developmental stages, disease stages, or different tissues facilitates deciphering of complex regulation network of cell differentiation, developmental control, and disease etiology and progression. We have developed a Bioconductor package called *REDSeq* to address the fundamental upstream analysis tasks of REDseq dataset. We have implemented functions for building genomic map of restriction enzyme sites (`buildREmap`), assigning sequencing tags to

*julie.zhu@umassmed.edu

RE sites (`assignSeq2REsite`), visualizing genome-wide distribution of differentially cut regions (`distanceHistSeq2RE`) and the distance distribution of sequence tags to corresponding RE sites (`distanceHistSeq2RE`), generating count table for identifying statistically significant RE sites (`summarizeByRE`). We have leveraged *BSgenome* on implementing function `buildREmap` for building genome-wide RE maps. The input data for `assignSeq2REsite` are represented as GRanges, for efficiently associating sequences with RE sites. It first identifies RE sites that have mapped sequence tags around the cut position taking consideration of user-defined offset, sequence length and strand in the aligned sequences. The user-defined offset guards against imperfect sticky end repair and primer addition process. These RE sites are used as seeds for assigning the remaining tags depending on which of five strategies the users select for partitioning sequences associated with multiple RE sites, i.e., unique, average, estimate, best and random. For experiment with at least two conditions with biological replicates, count summary generated from `summarizeByRE` can be easily used for identifying differentially cut RE sites using either *DESeq* or *edgeR*. Differentially cut RE sites can be annotated to the nearest gene using *ChIPpeakAnno*. In addition, for early stage experiments without replicates, `compareREDseq` outputs differentially cut RE sites between two experimental conditions using Fisher's Exact Test. For experiment with one experimental condition, `binom.test.REDseq` outputs differentially cut RE sites in the genome. Multiplicity adjustment functions from *multtest* package were integrated in both functions.

2 Examples of using REDseq

2.1 Task 1: Build a RE map for a genome

Given a fasta/fastq file containing the restriction enzyme recognition site and a BSgenome object, the function `buildREmap` builds a genome-wide RE map.

```
>         library(REDseq)
>         REpatternFilePath = system.file("extdata", "examplePattern.fa", package="REDseq")
>         library(BSgenome.Celegans.UCSC.ce2)
>         myMap = buildREmap( REpatternFilePath, BSgenomeName=Celegans, outfile="example.REmap")
```

```
>>> Finding all hits in sequences chrI ...
>>> DONE searching
>>> Finding all hits in sequences chrII ...
>>> DONE searching
>>> Finding all hits in sequences chrIII ...
>>> DONE searching
>>> Finding all hits in sequences chrIV ...
>>> DONE searching
>>> Finding all hits in sequences chrV ...
>>> DONE searching
>>> Finding all hits in sequences chrX ...
>>> DONE searching
>>> Finding all hits in sequences chrM ...
>>> DONE searching
```

2.2 Task 2: Assign mapped sequence tags to RE site

Given a mapped sequence tags as a GRanges and REmap as a Granges, `assignSeq2REsite` function assigns mapped sequence tags to RE site depending on the strategy users select. There are five strategies implemented, i.e., unique, average, estimate, best and random. For details, type help(`assignSeq2REsite`) in a R session.

```
>         data(example.REDseq)
>         data(example.map)
>         r.unique = assignSeq2REsite(example.REDseq, example.map, cut.offset = 1,
+         seq.length = 36, allowed.offset = 5, min.FragmentLength = 60,
+         max.FragmentLength = 300, partitionMultipleRE = "unique")

Tue Apr 25 18:07:50 2023 Validating input ...
Tue Apr 25 18:07:50 2023 Prepare map data ...
Tue Apr 25 18:07:50 2023 Align to chromosome  2  ...
Tue Apr 25 18:07:50 2023 Finished 1st round of aligning! Start the 2nd round of aligning ...
Tue Apr 25 18:07:50 2023 Align to chromosome  2  ...
Tue Apr 25 18:07:50 2023 Start filtering ...

>         r.best= assignSeq2REsite(example.REDseq, example.map,
+         cut.offset = 1, seq.length = 36, allowed.offset = 5,
+         min.FragmentLength = 60, max.FragmentLength = 300, partitionMultipleRE = "best")

Tue Apr 25 18:07:50 2023 Validating input ...
Tue Apr 25 18:07:50 2023 Prepare map data ...
Tue Apr 25 18:07:50 2023 Align to chromosome  2  ...
Tue Apr 25 18:07:50 2023 Finished 1st round of aligning! Start the 2nd round of aligning ...
Tue Apr 25 18:07:50 2023 Align to chromosome  2  ...
Tue Apr 25 18:07:50 2023 Start filtering ...
Tue Apr 25 18:07:50 2023 Partitioning reads over RE sites within  300 ...
Tue Apr 25 18:07:50 2023 get count for each RE ...

>         r.random = assignSeq2REsite(example.REDseq, example.map, cut.offset = 1,
+         seq.length = 36, allowed.offset = 5, min.FragmentLength = 60,
+         max.FragmentLength = 300, partitionMultipleRE = "random")

Tue Apr 25 18:07:50 2023 Validating input ...
Tue Apr 25 18:07:50 2023 Prepare map data ...
Tue Apr 25 18:07:50 2023 Align to chromosome  2  ...
Tue Apr 25 18:07:50 2023 Finished 1st round of aligning! Start the 2nd round of aligning ...
Tue Apr 25 18:07:50 2023 Align to chromosome  2  ...
Tue Apr 25 18:07:50 2023 Start filtering ...
Tue Apr 25 18:07:50 2023 Partitioning reads over RE sites within  300 ...

>         r.average = assignSeq2REsite(example.REDseq, example.map, cut.offset = 1,
+         seq.length = 36, allowed.offset = 5, min.FragmentLength = 60,
+          max.FragmentLength = 300, partitionMultipleRE = "average")

Tue Apr 25 18:07:50 2023 Validating input ...
Tue Apr 25 18:07:50 2023 Prepare map data ...
Tue Apr 25 18:07:50 2023 Align to chromosome  2  ...
Tue Apr 25 18:07:50 2023 Finished 1st round of aligning! Start the 2nd round of aligning ...
Tue Apr 25 18:07:50 2023 Align to chromosome  2  ...
Tue Apr 25 18:07:51 2023 Start filtering ...
Tue Apr 25 18:07:51 2023 Partitioning reads over RE sites within  300 ...

>         r.estimate = assignSeq2REsite(example.REDseq, example.map, cut.offset = 1,
+         seq.length = 36, allowed.offset = 5, min.FragmentLength = 60,
+         max.FragmentLength = 300, partitionMultipleRE = "estimate")
```

```
Tue Apr 25 18:07:51 2023 Validating input ...
Tue Apr 25 18:07:51 2023 Prepare map data ...
Tue Apr 25 18:07:51 2023 Align to chromosome  2  ...
Tue Apr 25 18:07:51 2023 Finished 1st round of aligning! Start the 2nd round of aligning ...
Tue Apr 25 18:07:51 2023 Align to chromosome  2  ...
Tue Apr 25 18:07:51 2023 Start filtering ...
Tue Apr 25 18:07:51 2023 Partitioning reads over RE sites within  300 ...
Tue Apr 25 18:07:51 2023 get count for each RE ...
```

```
> head(r.estimate$passed.filter)
```

```
      SEQid            REid Chr strand SEQstart  SEQend REstart   REend
2  00000037 Sau96I.chr10.29   2      +  3012091 3012126 3012090 3012094
3  00000038 Sau96I.chr10.29   2      +  3012096 3012131 3012090 3012094
5  00000040 Sau96I.chr10.30   2      +  3012300 3012335 3012299 3012303
8  00000055 Sau96I.chr10.42   2      +  3018315 3018350 3018314 3018318
9  00000056 Sau96I.chr10.42   2      +  3018315 3018350 3018314 3018318
10 00000057 Sau96I.chr10.42   2      +  3018315 3018350 3018314 3018318
   Distance Weight
2         1      1
3         6      1
5         1      1
8         1      1
9         1      1
10        1      1
```

The above examples are for single-end sequencing data. For paired-end sequencing data, please create inputS.RD and inputE.RD from input.RD first with start(input.RD) and end(input.RD), where inputS.RD contains the start of the input.RD and inputE.RD contains the end of the input.RD. Then call assignSeq2REsite twice with inputS.RD and inputE.RD respectively. Please set min.FragmentLength = 0, max.FragmentLength = 1, seq.length = 1 with both calls.

2.3 Task 3: Visualize the distribution of cut frequency in selected genomic regions and the distance distribution of sequence tags to corresponding RE sites

```
> data(example.assignedREDseq)
```

```
> plotCutDistribution(example.assignedREDseq,example.map, chr="2",
+ xlim =c(3012000, 3020000))
```

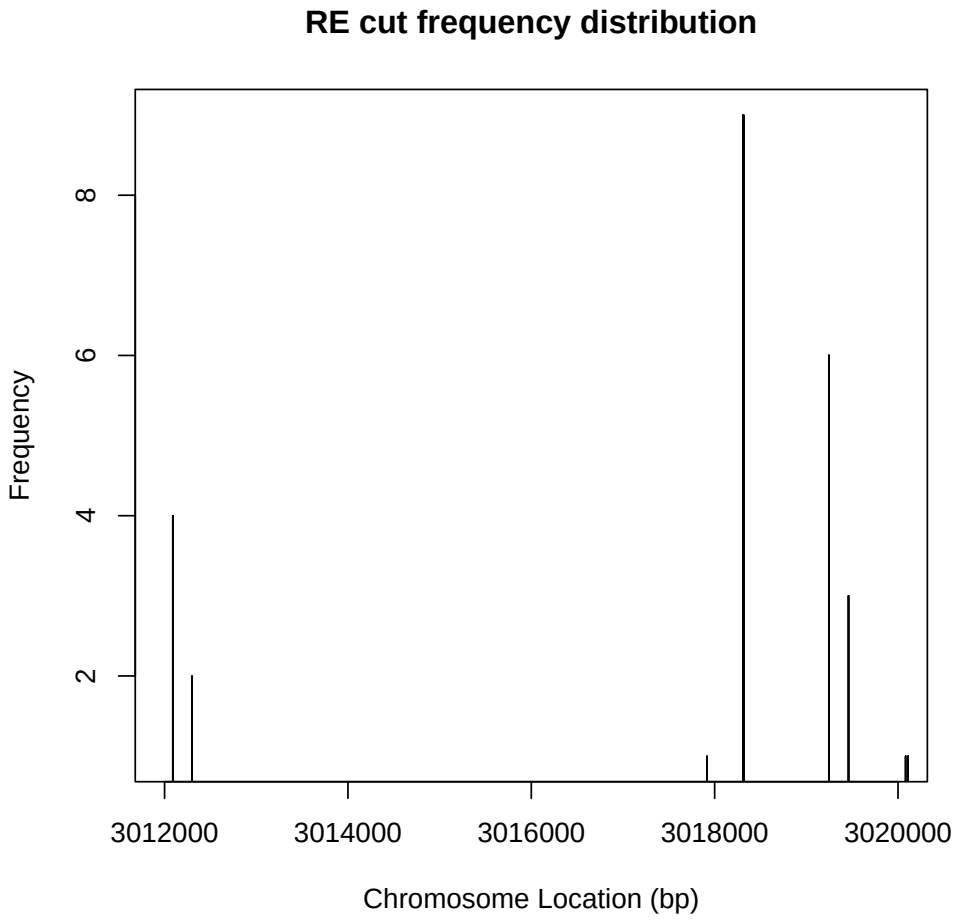

Figure 1: Plot to show the distribution of cut frequency in the selected genomic-regions with the function `plotCutDistribution`. The red triangle is the expected cut frequency for each RE site.

```
> distanceHistSeq2RE(example.assignedREDseq,ylim=c(0,25))
```

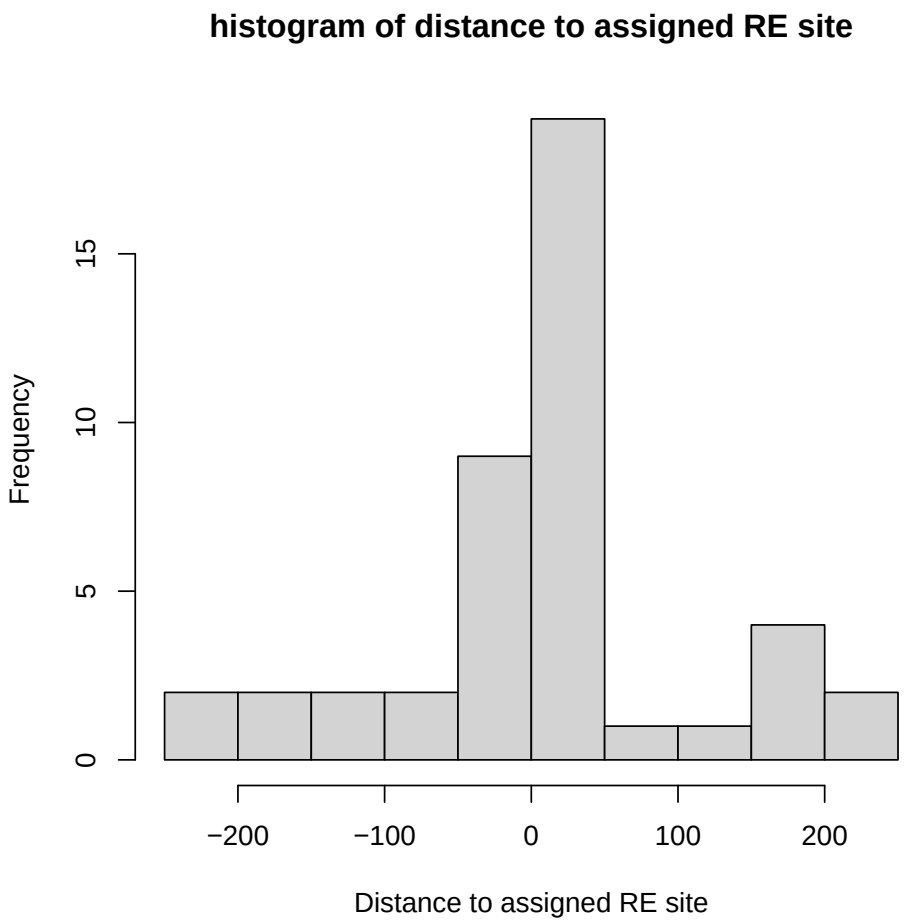

Figure 2: Plot to show the distribution of distance of sequence tags to associated RE sites with the function `distanceHistSeq2RE`.

2.4 Task 4: Generating count table for identifying statistically significant RE sites

Once you have obtained the assigned RE sites, you can use the function `summarizeByRE` to obtain a count table for identifying statistically significant RE sites using *DEseq* or *edgeR*.

```
> REsummary  =summarizeByRE(example.assignedREDseq,by="Weight")
```

2.5 Task 5: Identifying differential cut RE sites for experiment with one experiment condition

```
> binom.test.REDseq(REsummary)
```

```
        p.value total.weight.count            REid cut.frequency
1  2.804822e-47                 9 Sau96I.chr10.42       0.28125
2  9.061718e-31                 6 Sau96I.chr10.43       0.18750
3  3.595919e-20                 4 Sau96I.chr10.29       0.12500
4  4.959892e-15                 3 Sau96I.chr10.50       0.09375
5  4.959892e-15                 3 Sau96I.chr10.45       0.09375
6  4.959901e-10                 2 Sau96I.chr10.30       0.06250
7  4.959901e-10                 2 Sau96I.chr10.51       0.06250
8  3.199950e-05                 1 Sau96I.chr10.40       0.03125
9  3.199950e-05                 1 Sau96I.chr10.49       0.03125
10 3.199950e-05                 1 Sau96I.chr10.47       0.03125
   BH.adjusted.p.value
1         2.804822e-46
2         4.530859e-30
3         1.198640e-19
4         9.919784e-15
5         9.919784e-15
6         7.085573e-10
7         7.085573e-10
8         3.199950e-05
9         3.199950e-05
10        3.199950e-05
```

2.6 Task 6: Identifying differential cut RE sites for early stage experiment without replicates

```
> x= cbind(c("RE1", "RE2", "RE3", "RE4"), c(10,1,100, 0),c(5,5,50, 40))
> colnames(x) = c("REid", "control", "treated")
> compareREDseq(x)
```

```
       p.value control.count treated.count REid control.total treated.total
1 6.233642e-16             0            40  RE4           111           100
2 1.159388e-10           100            50  RE3           111           100
3 1.035503e-01             1             5  RE2           111           100
4 2.943364e-01            10             5  RE1           111           100
  odds.ratio BH.adjusted.p.value
1        Inf        2.493457e-15
2  0.1112945        2.318777e-10
3  5.7478720        1.380671e-01
4  0.5331227        2.943364e-01
```

3 References

1. Roberts, R.J., Restriction endonucleases. CRC Crit Rev Biochem, 1976. 4(2): p. 123-64.

2. Kessler, C. and V. Manta, Specificity of restriction endonucleases and DNA modification methyltransferases a review (Edition 3). Gene, 1990. 92(1-2): p. 1-248.

3. Pingoud, A., J. Alves, and R. Geiger, Restriction enzymes. Methods Mol Biol, 1993. 16: p. 107-200.

4. bibitemAnders10 Anders, S. and W. Huber, Differential expression analysis for sequence count data. Genome Biol, 2010. 11(10): p. R106.

5. Robinson, M.D., D.J. McCarthy, and G.K. Smyth, edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics, 2010. 26(1): p. 139-40.

6. Zhu, L.J., et al., ChIPpeakAnno: a Bioconductor package to annotate ChIP-seq and ChIP-chip data. BMC Bioinformatics, 2010. 11: p. 237.

7. Pages, H., BSgenome package. http://bioconductor.org/packages/2.8/bioc/vignettes/BSgenome/inst/doc/GenomeSearching.pdf

8. Zhu, L.J., et al., REDseq: A Bioconductor package for Analyzing High Throughput Sequencing Data from Restriction Enzyme Digestion. (In preparation)

4 Session Info

```
> sessionInfo()

R version 4.3.0 RC (2023-04-18 r84287)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 22.04.2 LTS

Matrix products: default
BLAS:   /home/biocbuild/bbs-3.18-bioc/R/lib/libRblas.so
LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.10.0

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_GB              LC_COLLATE=C
 [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

time zone: America/New_York
tzcode source: system (glibc)

attached base packages:
[1] stats4    stats     graphics  grDevices utils     datasets  methods
[8] base

other attached packages:
 [1] REDseq_1.47.0                    ChIPpeakAnno_3.35.0
 [3] multtest_2.57.0                  Biobase_2.61.0
 [5] BSgenome.Celegans.UCSC.ce2_1.4.0 BSgenome_1.69.0
 [7] rtracklayer_1.61.0               Biostrings_2.69.0
 [9] XVector_0.41.0                   GenomicRanges_1.53.0
[11] GenomeInfoDb_1.37.0              IRanges_2.35.0
[13] S4Vectors_0.39.0                 BiocGenerics_0.47.0

loaded via a namespace (and not attached):
 [1] tidyselect_1.2.0            dplyr_1.1.2
 [3] blob_1.2.4                  filelock_1.0.2
 [5] bitops_1.0-7                fastmap_1.1.1
 [7] RCurl_1.98-1.12             BiocFileCache_2.9.0
 [9] GenomicAlignments_1.37.0    XML_3.99-0.14
[11] digest_0.6.31               lifecycle_1.0.3
[13] survival_3.5-5              KEGGREST_1.41.0
[15] VennDiagram_1.7.3           RSQLite_2.3.1
[17] magrittr_2.0.3              compiler_4.3.0
[19] rlang_1.1.0                 progress_1.2.2
[21] tools_4.3.0                 utf8_1.2.3
[23] yaml_2.3.7                  lambda.r_1.2.4
[25] prettyunits_1.1.1           bit_4.0.5
[27] curl_5.0.0                  DelayedArray_0.27.0
[29] regioneR_1.33.0             xml2_1.3.3
[31] BiocParallel_1.35.0         grid_4.3.0
[33] fansi_1.0.4                 colorspace_2.1-0
[35] ggplot2_3.4.2               scales_1.2.1
[37] MASS_7.3-59                 biomaRt_2.57.0
[39] SummarizedExperiment_1.31.0 cli_3.6.1
[41] crayon_1.5.2                generics_0.1.3
[43] httr_1.4.5                  rjson_0.2.21
[45] DBI_1.1.3                   cachem_1.0.7
```

```
[47] stringr_1.5.0              zlibbioc_1.47.0
[49] splines_4.3.0              parallel_4.3.0
[51] AnnotationDbi_1.63.0       formatR_1.14
[53] restfulr_0.0.15            matrixStats_0.63.0
[55] vctrs_0.6.2                Matrix_1.5-4
[57] hms_1.1.3                  bit64_4.0.5
[59] RBGL_1.77.0                GenomicFeatures_1.53.0
[61] glue_1.6.2                 codetools_0.2-19
[63] gtable_0.3.3               stringi_1.7.12
[65] futile.logger_1.4.3        BiocIO_1.11.0
[67] munsell_0.5.0              tibble_3.2.1
[69] pillar_1.9.0               rappdirs_0.3.3
[71] graph_1.79.0               GenomeInfoDbData_1.2.10
[73] R6_2.5.1                   dbplyr_2.3.2
[75] lattice_0.21-8             futile.options_1.0.1
[77] png_0.1-8                  Rsamtools_2.17.0
[79] memoise_2.0.1              Rcpp_1.0.10
[81] InteractionSet_1.29.0      MatrixGenerics_1.13.0
[83] pkgconfig_2.0.3
```