The REDseq user's guide

Lihua Julie Zhu*

April 30, 2018

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 RangedData, 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 RangedData and REmap as a RangedData, `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")

Mon Apr 30 21:33:30 2018 Validating input ...
Mon Apr 30 21:33:30 2018 Prepare map data ...
Mon Apr 30 21:33:30 2018 Align to chromosome  2  ...
Mon Apr 30 21:33:30 2018 Finished 1st round of aligning! Start the 2nd round of aligning ...
Mon Apr 30 21:33:30 2018 Align to chromosome  2  ...
Mon Apr 30 21:33:30 2018 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")

Mon Apr 30 21:33:30 2018 Validating input ...
Mon Apr 30 21:33:30 2018 Prepare map data ...
Mon Apr 30 21:33:30 2018 Align to chromosome  2  ...
Mon Apr 30 21:33:30 2018 Finished 1st round of aligning! Start the 2nd round of aligning ...
Mon Apr 30 21:33:30 2018 Align to chromosome  2  ...
Mon Apr 30 21:33:30 2018 Start filtering ...
Mon Apr 30 21:33:30 2018 Partitioning reads over RE sites within  300 ...
Mon Apr 30 21:33:30 2018 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")

Mon Apr 30 21:33:30 2018 Validating input ...
Mon Apr 30 21:33:30 2018 Prepare map data ...
Mon Apr 30 21:33:30 2018 Align to chromosome  2  ...
Mon Apr 30 21:33:30 2018 Finished 1st round of aligning! Start the 2nd round of aligning ...
Mon Apr 30 21:33:30 2018 Align to chromosome  2  ...
Mon Apr 30 21:33:30 2018 Start filtering ...
Mon Apr 30 21:33:30 2018 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")

Mon Apr 30 21:33:30 2018 Validating input ...
Mon Apr 30 21:33:30 2018 Prepare map data ...
Mon Apr 30 21:33:30 2018 Align to chromosome  2  ...
Mon Apr 30 21:33:30 2018 Finished 1st round of aligning! Start the 2nd round of aligning ...
Mon Apr 30 21:33:30 2018 Align to chromosome  2  ...
Mon Apr 30 21:33:30 2018 Start filtering ...
Mon Apr 30 21:33:30 2018 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")
```

```
Mon Apr 30 21:33:30 2018 Validating input ...
Mon Apr 30 21:33:30 2018 Prepare map data ...
Mon Apr 30 21:33:30 2018 Align to chromosome  2  ...
Mon Apr 30 21:33:30 2018 Finished 1st round of aligning! Start the 2nd round of aligning ...
Mon Apr 30 21:33:30 2018 Align to chromosome  2  ...
Mon Apr 30 21:33:30 2018 Start filtering ...
Mon Apr 30 21:33:30 2018 Partitioning reads over RE sites within  300 ...
Mon Apr 30 21:33:30 2018 get count for each RE ...
```

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

```
     SEQid            REid Chr strand SEQstart  SEQend REstart   REend Distance
1 00000036 Sau96I.chr10.29   2     -1  3012058 3012093 3012090 3012094      -32
2 00000037 Sau96I.chr10.29   2      1  3012091 3012126 3012090 3012094        1
3 00000038 Sau96I.chr10.29   2      1  3012096 3012131 3012090 3012094        6
4 00000039 Sau96I.chr10.30   2     -1  3012266 3012301 3012299 3012303      -33
5 00000040 Sau96I.chr10.30   2      1  3012300 3012335 3012299 3012303        1
6 00000052 Sau96I.chr10.40   2     -1  3017881 3017916 3017916 3017920      -35
  Weight
1      1
2      1
3      1
4      1
5      1
6      1
```

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))
```

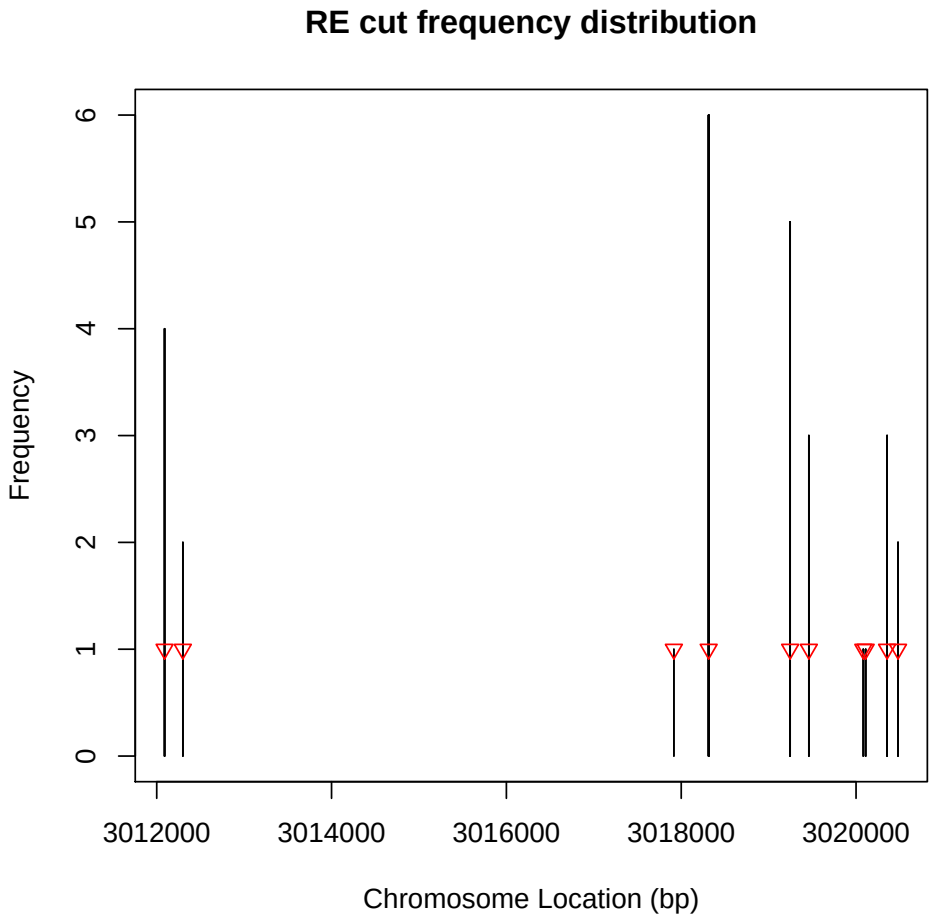

Figure 1: Plot to show the distribution of cut frequency in the selected genomic-regions with the function `plotCutDistribution`. The red triangle is the expected cut frequency for each RE site.

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

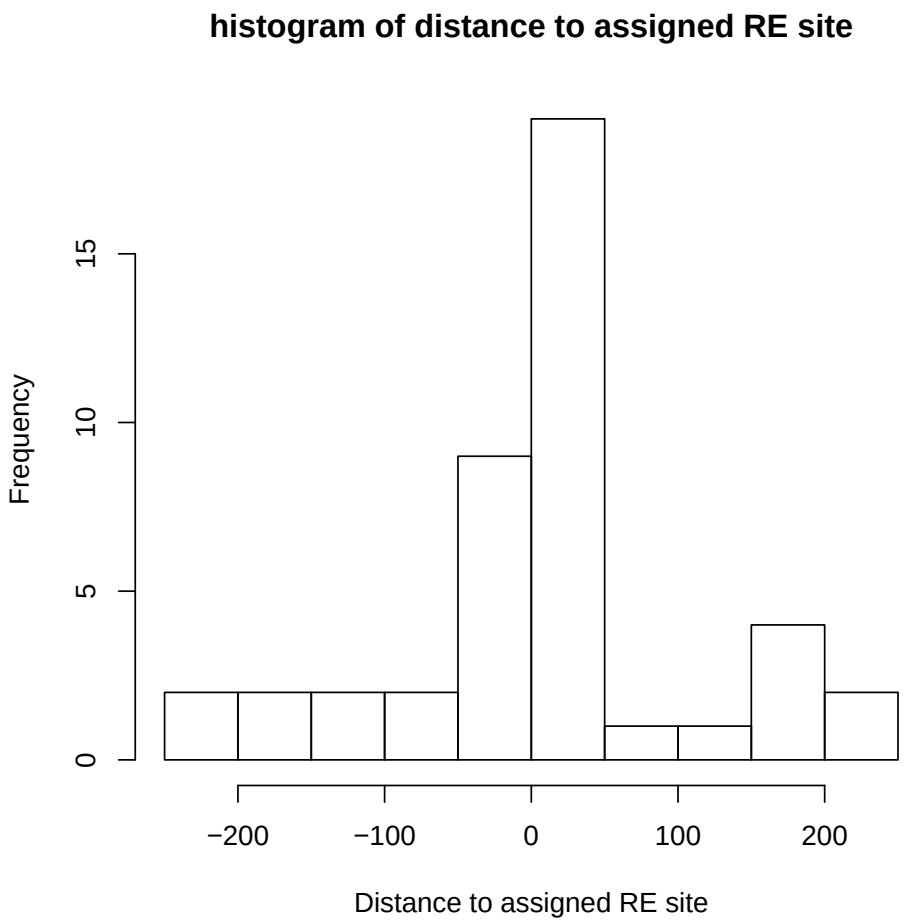

Figure 2: Plot to show the distribution of distance of sequence tags to associated RE sites with the function `distanceHistSeq2RE`.

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 3.5.0 (2018-04-23)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows Server 2012 R2 x64 (build 9600)

Matrix products: default

locale:
[1] LC_COLLATE=C
[2] LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.1252

attached base packages:
```

```
 [1] grid      stats4    parallel  stats     graphics  grDevices utils
 [8] datasets  methods   base

other attached packages:
 [1] REDseq_1.26.0                    ChIPpeakAnno_3.14.0
 [3] VennDiagram_1.6.20               futile.logger_1.4.3
 [5] multtest_2.36.0                  Biobase_2.40.0
 [7] BSgenome.Celegans.UCSC.ce2_1.4.0 BSgenome_1.48.0
 [9] rtracklayer_1.40.0               Biostrings_2.48.0
[11] XVector_0.20.0                   GenomicRanges_1.32.0
[13] GenomeInfoDb_1.16.0              IRanges_2.14.0
[15] S4Vectors_0.18.0                 BiocGenerics_0.26.0

loaded via a namespace (and not attached):
 [1] Rcpp_0.12.16                lattice_0.20-35
 [3] GO.db_3.6.0                 prettyunits_1.0.2
 [5] Rsamtools_1.32.0            assertthat_0.2.0
 [7] digest_0.6.15               R6_2.2.2
 [9] futile.options_1.0.1        RSQLite_2.1.0
[11] httr_1.3.1                  BiocInstaller_1.30.0
[13] zlibbioc_1.26.0             GenomicFeatures_1.32.0
[15] progress_1.1.2              lazyeval_0.2.1
[17] curl_3.2                    blob_1.1.1
[19] Matrix_1.2-14               splines_3.5.0
[21] BiocParallel_1.14.0         stringr_1.3.0
[23] ProtGenerics_1.12.0         RCurl_1.95-4.10
[25] bit_1.1-12                  biomaRt_2.36.0
[27] DelayedArray_0.6.0          compiler_3.5.0
[29] pkgconfig_2.0.1             SummarizedExperiment_1.10.0
[31] GenomeInfoDbData_1.1.0      idr_1.2
[33] matrixStats_0.53.1          XML_3.98-1.11
[35] GenomicAlignments_1.16.0    MASS_7.3-50
[37] bitops_1.0-6                RBGL_1.56.0
[39] DBI_0.8                     AnnotationFilter_1.4.0
[41] magrittr_1.5                formatR_1.5
[43] graph_1.58.0                stringi_1.1.7
[45] limma_3.36.0                seqinr_3.4-5
[47] lambda.r_1.2.2              ensembldb_2.4.0
[49] tools_3.5.0                 ade4_1.7-11
[51] bit64_0.9-7                 survival_2.42-3
[53] AnnotationDbi_1.42.0        regioneR_1.12.0
[55] memoise_1.1.0
```