Package ‘REDseq’

January 31, 2017

Type Package

Title Analysis of high-throughput sequencing data processed by restriction enzyme digestion

Version 1.20.0

Date 2013-10-21

Author Lihua Julie Zhu and Thomas Fazzio

Maintainer Lihua Julie Zhu <julie.zhu@umassmed.edu>

Depends R (>= 2.15.0), BiocGenerics (>= 0.1.0), BSgenome.Celegans.UCSC.ce2, multtest, Biostrings, BSgenome, ChIPpeakAnno

Imports BiocGenerics, AnnotationDbi, Biostrings, ChIPpeakAnno, graphics, IRanges (>= 1.13.5), multtest, stats, utils

biocViews Sequencing, SequenceMatching, Preprocessing

Description The package includes functions to build restriction enzyme cut site (RECS) map, distribute mapped sequences on the map with five different approaches, find enriched/depleted RECSs for a sample, and identify differentially enriched/depleted RECSs between samples.

License GPL (>=2)

NeedsCompilation no

R topics documented:

REDseq-package ...................................................... 2
assignSeq2REsite .................................................. 3
binom.test.REDseq .................................................. 5
buildREmap .......................................................... 6
compareREDseq ..................................................... 7
distanceHistSeq2RE ............................................... 8
example.assignedREDseq .......................................... 9
example.map ......................................................... 10
example.REDseq ................................................... 10
plotCutDistribution ............................................... 11
summarizeByRE .................................................... 12
summarizeBySeq ................................................... 13

Index 14
Description

REDSeq is a Bioconductor package for building genomic map of restriction enzyme sites REmap, assigning sequencing tags to RE sites using five different strategies, visualizing genome-wide distribution of differentially cut regions with the REmap as reference and the distance distribution of sequence tags to corresponding RE sites, generating count table for identifying statistically significant RE sites using edgeR or DEseq.

Details

<table>
<thead>
<tr>
<th>Package:</th>
<th>REDseq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Package</td>
</tr>
<tr>
<td>Version:</td>
<td>1.0</td>
</tr>
<tr>
<td>Date:</td>
<td>2011-05-10</td>
</tr>
<tr>
<td>License:</td>
<td>GPL</td>
</tr>
<tr>
<td>LazyLoad:</td>
<td>yes</td>
</tr>
</tbody>
</table>

~~ An overview of how to use the package, including the most important functions ~~

Author(s)

Lihua Julie Zhu

Maintainer: Lihua Julie Zhu <julie.zhu@umassmed.edu>

References

See Also

buildREmap, assignSeq2REsite, plotCutDistribution, distanceHistSeq2RE, summarizeByRE, summarizeBySeq, compareREseq, binom.test.REDseq

Examples

if(interactive()){
library(ChIPpeakAnno)
REpatternFilePath = system.file("extdata", "examplePattern.fa", package="REDseq")
library(BSgenome.Celegans.UCSC.ce2)
buildREmap(REpatternFilePath, BSgenomeName=Celegans, outfile=tempfile())
library(REDseq)
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")
r.average = assignSeq2REsite(example.REDseq, example.map, cut.offset = 1,
seq.length = 36, allowed.offset = 5, min.FragmentLength = 60,
max.FragmentLength = 300, partitionMultipleRE = "average")
r.random = assignSeq2REsite(example.REDseq, example.map, cut.offset = 1,
seq.length = 36, allowed.offset = 5, min.FragmentLength = 60,
max.FragmentLength = 300, partitionMultipleRE = "random")
r.best = assignSeq2REsite(example.REDseq, example.map, cut.offset = 1,
seq.length = 36, allowed.offset = 5, min.FragmentLength = 60,
max.FragmentLength = 300, partitionMultipleRE = "best")
r.estimate = assignSeq2REsite(example.REDseq, example.map, cut.offset = 1,
seq.length = 36, allowed.offset = 5, min.FragmentLength = 60,
max.FragmentLength = 300, partitionMultipleRE = "estimate")
r.estimate$passed.filter
data(example.assignedREDseq)
plotCutDistribution(example.assignedREDseq, example.map,
chr="2", xlim =c(3012000, 3020000))
distanceHistSeq2RE(example.assignedREDseq, ylim=c(0,20))
summarizeByRE(example.assignedREDseq, by="Weight", sampleName="example")
REsummary =summarizeByRE(example.assignedREDseq, by="Weight")
binom.test.REDseq(REsummary)
}

assignSeq2REsite

Assign mapped sequence tags to corresponding restriction enzyme (RE) cut sites

Description

Given the sequence tags aligned to a genome as a RangedData, and a map built using the buildREmap function, assignSeq2REsite 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. 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.
Usage

assignSeq2REsite(input.RD, REmap.RD, cut.offset = 1, seq.length = 36, allowed.offset = 5, min.FragmentLength = 60, max.FragmentLength = 300, partitionMultipleRE = c("unique", "average", "estimate", "best", "random"))

Arguments

- **input.RD**: RangedData as mapped sequences: see example below
- **REmap.RD**: RangedData as restriction enzyme (RE) cut site map: see example below
- **cut.offset**: The cut offset from the start of the RE recognition sequence: index is 0 based, i.e., 1 means the RE cuts at position 2.
- **seq.length**: Sequence length: 36 means that the sequence tags are 36-base long.
- **allowed.offset**: Offset allowed to count for imperfect sticky end repair and primer addition.
- **min.FragmentLength**: Minimum fragment length of the sequences size-selected for sequencing
- **max.FragmentLength**: Maximum fragment length of the sequences size-selected for sequencing
- **partitionMultipleRE**: The strategy for partitioning sequences associated with multiple RE sites. For strategy unique, only sequence tags that are associated with a unique RE site within the distance between min.FragmentLength and max.FragmentLength are kept for downstream analysis. For strategy average, sequence tags are partitioned equally among associated RE sites. For strategy estimate, sequence tags are partitioned among associated RE sites with a weight function, which is determined using the count distribution of the RE seed sites described in the description section above. For strategy best, sequence tags are assigned to the most probable RE sites with the same weight function as that in strategy estimate. For strategy random, the sequence tags are randomly assigned to one of the multiple associated RE sites.

Value

- **passed.filter**: Sequences assigned to RE(s), see the example r.unique$passed.filter
- **notpassed.filter**: Sequences not assigned to any RE, see example r.unique$notpassed.filter
- **mREwithDetail**: Detailed assignment information for sequences associated with multiple RE sites. Only available when partitionMultipleRE is set to average or estimate, see r.estimate$mREwithDetail in the examples

Author(s)

Lihua Julie Zhu

References

binom.test.REDseq

**Binomial test for REDseq dataset**

**Description**
For any early stage experiment with one experimental condition and one biological replicate, binom.test.REDseq computes p-value for each RE site in the genome.

**Usage**

```
binom.test.REDseq(REsummary, col.count = 2, multiAdj = TRUE, multiAdjMethod = "BH", prior.p = 0.000001)
```

**Arguments**

- **REsummary**: A matrix returned from summarizeByRE with a RE id column, a count/weight column. See examples
- **col.count**: The column where the total count/weight is
- **multiAdj**: Whether apply multiple hypothesis testing adjustment, TRUE or FALSE
- **multiAdjMethod**: Multiple testing procedures, for details, see mt.rawp2adjp in multtest package
- **prior.p**: It is the probability of assigning a mapped sequence tag to a given RE site. Assuming each RE site gets cut equally, then the prior.p = 1/number of total RE sites in the genome.

**Examples**

```r
library(REDseq)
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")
r.average = assignSeq2REsite(example.REDseq, example.map, cut.offset = 1, seq.length = 36, allowed.offset = 5, min.FragmentLength = 60, max.FragmentLength = 300, partitionMultipleRE = "average")
r.random = assignSeq2REsite(example.REDseq, example.map, cut.offset = 1, seq.length = 36, allowed.offset = 5, min.FragmentLength = 60, max.FragmentLength = 300, partitionMultipleRE = "random")
r.best = assignSeq2REsite(example.REDseq, example.map, cut.offset = 1, seq.length = 36, allowed.offset = 5, min.FragmentLength = 60, max.FragmentLength = 300, partitionMultipleRE = "best")
r.estimate = assignSeq2REsite(example.REDseq, example.map, cut.offset = 1, seq.length = 36, allowed.offset = 5, min.FragmentLength = 60, max.FragmentLength = 300, partitionMultipleRE = "estimate")
r.estimate$passed.filter
r.estimate$notpassed.filter
```
Value

- `p.value`: p-value of the test
- `*.count`: weight/count from the input `REsummary`
- `REid`: id of the restriction enzyme from the input `REsummary`
- `cut.frequency`: cut frequency
- `*.adjusted.p.value`: applicable if `multiAdj=TRUE`, adjusted p-value using * method specified in `multiAdjMethod`

Author(s)

Lihua Julie Zhu

See Also

`compareREDseq`

Examples

```r
library(REDseq)
REsummary = cbind(c("RE1", "RE2", "RE3"), c(10,1,100))
colnames(REsummary) = c("REid", "control")
binom.test.REDseq(REsummary)
```

---

**buildREmap**

*Build a genome wide cut site map for a Restriction Enzyme (RE)*

Description

Build a genome-wide cut map for a Restriction Enzyme (RE)

Usage

```r
buildREmap(REpatternFilePath, format = "fasta", BSgenomeName, outfile)
```

Arguments

- **`REpatternFilePath`**: File path storing the recognition pattern of a RE
- **`format`**: format of the pattern file, either "fasta" (the default) or "fastq"
- **`BSgenomeName`**: BSgenome object, please refer to available.genomes in BSgenome package for details
- **`outfile`**: temporary output file for writing the matched chromosome location to

Value

Output REmap as a RangedData

Author(s)

Lihua Julie Zhu
Examples

```r
library(REDseq)
REpatternFilePath = system.file("extdata", "examplePattern.fa", package="REDseq")
library(BSgenome.Celegans.UCSC.ce2)
buildREmap( REpatternFilePath, BSgenomeName=Celegans, outfile=tempfile())
```

Description

For early stage experiment without replicates, compareREDseq outputs differentially cut RE sites between two experimental conditions using Fisher’s Exact Test.

Usage

```r
compareREDseq(REsummary, col.count1 = 2, col.count2 = 3, multiAdj = TRUE,
             multiAdjMethod = "BH", maxP = 1, minCount = 1)
```

Arguments

- **REsummary**: A matrix with a RE id column, 2 count/weight column, see examples
- **col.count1**: The column where the total count/weight for the 1st experimental condition is
- **col.count2**: The column where the total count/weight for the 2nd experimental condition is
- **multiAdj**: Whether apply multiple hypothesis testing adjustment, TURE or FALSE
- **multiAdjMethod**: Multiple testing procedures, for details, see mt.rawp2adjp in multtest package
- **maxP**: The maximum p-value to be considered to be significant
- **minCount**: For a RE site to be included, the tag count from at least one of the experimental conditions >= minimumCount

Value

- **p.value**: the p-value of the test
- ***.count**: weight/count from the input column col.count1 and col.count2
- ***.total**: total weight/count from input column col.count1 and col.count2
- **REid**: the id of the restriction enzyme from the input
- **odds.ratio**: an estimate of the odds ratio for 2nd experimental condition vs. 1st experimental condition
- ***.adjusted.p.value**: applicable if multiAdj=TRUE, adjusted p.value using the method * specified in multiAdjMethod

Author(s)

Lihua Julie Zhu

See Also

binom.test.REDseq
Examples

library(REDseq)

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)

distanceHistSeq2RE

Plot the distance distribution from sequence to the associated RE sites

Description

Give an overview of the distance distribution from all assigned sequences to the associated RE sites. If average or estimate is used for assigning sequences to RE sites, the count for histogram drawing will be adjusted with the weight assigned.

Usage

distanceHistSeq2RE(assignedSeqs, longestDist = 1000,
                   title = "histogram of distance to assigned RE site",
                   xlab = "Distance to assigned RE site", ylab = "Frequency", ylim="")

Arguments

assignedSeqs     result returned from assignSeq2REsite
longestDist      longest distance to keep in the plot
title            an overall title for the plot
xlab             a title for the x axis
ylab             a title for the y axis
ylim             range of y to be plotted

Author(s)

Lihua Julie Zhu

See Also

assignSeq2REsite, distanceHistSeq2RE

Examples

library(REDseq)

data(example.assignedREDseq)

distanceHistSeq2RE(example.assignedREDseq, ylim=c(0,20))
Description

an example assigned REDseq dataset generated from assignSeq2REsite

Usage

data(example.assignedREDseq)

Format

The format is: List of 3
$ passed.filter :'data.frame': Sequences that passed the filters:
  ..$ SEQid :Sequence ID
  ..$ REid : Restriction Enzyme Site ID
  ..$ Chr : Chromosome
  ..$ strand : Strand
  ..$ SEQstart: Sequence Start
  ..$ SEQend : Sequence End
  ..$ REstart : Restriction Enzyme Site Start
  ..$ REend : Restriction Enzyme Site End
  ..$ Distance: Distance from SEQstart to REstart
  ..$ Weight : Weighted count for this REid and this SEQid
$ notpassed.filter:'data.frame': Sequences that did not pass the filters
  ..$ SEQid :Sequence ID
  ..$ REid : Restriction Enzyme Site ID
  ..$ Chr : Chromosome
  ..$ strand : Strand
  ..$ SEQstart: Sequence Start
  ..$ SEQend : Sequence End
  ..$ REstart : Restriction Enzyme Site Start
  ..$ REend : Restriction Enzyme Site End
  ..$ Distance: Distance from SEQstart to REstart
  ..$ Weight : Weighted count for this REid and this SEQid
$ mREwithDetail :'data.frame': Detailed information about the sequences that are associated with
  ..$ SEQid :Sequence ID
  ..$ REid : Restriction Enzyme Site ID
  ..$ Chr : Chromosome
  ..$ strand : Strand
  ..$ SEQstart: Sequence Start
  ..$ SEQend : Sequence End
  ..$ REstart : Restriction Enzyme Site Start
  ..$ REend : Restriction Enzyme Site End
  ..$ Distance: Distance from SEQstart to REstart
  ..$ Weight : Weighted count for this REid and this SEQid
  ..$ count : count of seed for this REid and SEQid
Example.

..S total.count: total number of seeds that are associated with this SEQId

**Examples**

```r
library(REDseq)
data(example.assignedREDseq)
## maybe str(example.assignedREDseq) ; plot(example.assignedREDseq) ...
```

---

**example.map**  
*an example REmap dataset*

**Description**

an example REmap dataset as RangedData generated from buildREmap

**Usage**

```r
data(example.map)
```

**Format**

The format is: Formal class 'RangedData' [package "IRanges"]

**Examples**

```r
library(REDseq)
data(example.map)
## maybe str(example.map) ; plot(example.map) ...
```

---

**example.REDseq**  
*an example sequencing dataset from a restoration enzyme digestion (RED) experiment*

**Description**

an example RED sequencing dataset as a RangedData

**Usage**

```r
data(example.REDseq)
```

**Format**

The format is: Formal class 'RangedData' [package "IRanges"]

**Examples**

```r
library(REDseq)
data(example.REDseq)
## maybe str(example.REDseq) ; plot(example.REDseq) ...
```
plotCutDistribution

plot cut frequencies of RE sites along a given chromosome

Description

plot cut frequencies of RE sites along a chromosome, which gives a bird-eye view of genome-wide frequent-cut regions and RE inaccessible regions.

Usage

plotCutDistribution(assignedSeqs, REmap, chr="chr1", xlim, title="RE cut frequency distribution", xlab="Chromosome Location (bp)", ylab="Frequency", round=TRUE, n.sequence)

Arguments

- assignedSeqs: result returned from assignSeq2REsite
- REmap: REmap used in assignSeq2REsite and generated from buildREmap
- chr: chromosome to be plotted
- xlim: range of x to be plotted
- title: an overall title for the plot
- xlab: a title for the x axis
- ylab: a title for the y axis
- round: TRUE: the sum of the weight is rounded up if the fraction part is greater than 0.5. FALSE: as it is.
- n.sequence: total uniquely mapped sequences in the dataset for estimating the expected count for each RE site. If omitted, the expected count for each RE site will be set as 1 as default.

Author(s)

Lihua Julie Zhu

See Also

assignSeq2REsite, distanceHistSeq2RE

Examples

library(REDseq)
data(example.assignedREDseq)
data(example.map)
plotCutDistribution(example.assignedREDseq, example.map, chr="2", xlim = c(3012000, 3020000))
summarizeByRE

Output count/weight summary by restriction enzyme cut site ID (REid)

Description

Output count/weight summary by REid with each row representing each REid

Usage

summarizeByRE(assignedSeqs, by=c("Weight", "REid"), sampleName="", round=TRUE)

Arguments

assignedSeqs  output from assignSeq2REsite
by             Weight if sum up the weight for each REid, REid if sum the occurrence of each REid.
sampleName     The name of the sample used as the count column name.
round           TRUE: the sum of the weight is rounded up if the fraction part is greater than 0.5. FALSE: as it is.

Value

a matrix with REid as the first column and total count/weight as the second column, that can be used for the downstream analysis with DEseq or edgeR.

Author(s)

Lihua Julie Zhu

See Also

summarizeBySeq, assignSeq2REsite

Examples

library(REDseq)
data(example.assignedREDseq)
summarizeByRE(example.assignedREDseq, by="REid", sampleName="example")
summarizeByRE(example.assignedREDseq, by="Weight", sampleName="example")
**summarizeBySeq**

Output count/weight summary by sequences

**Description**
Output count/weight summary by sequences with each row representing each sequence.

**Usage**
```
summarizeBySeq(assignedSeqs, by=c("Weight", "SEQid"))
```

**Arguments**
- `assignedSeqs`: output from `assignSeq2REsite`
- `by`: `Weight` if sum up the weight for each sequence, `SEQid` if sum the occurrence of each sequence

**Value**
a matrix with `SEQid` as the first column and total count/weight as the second column

**Author(s)**
Lihua Julie Zhu

**See Also**
`summarizeByRE`, `assignSeq2REsite`

**Examples**
```
library(REDseq)
data(example.assignedREDseq)
summarizeBySeq(example.assignedREDseq, by="Weight")
summarizeBySeq(example.assignedREDseq, by="SEQid")
```
Index

*Topic **Statistics**
  - binom.test.REDseq, 5
  - compareREDseq, 7
*Topic **datasets**
  - example.assignedREDseq, 9
  - example.map, 10
  - example.REDseq, 10
*Topic **graph**
  - distanceHistSeq2RE, 8
  - plotCutDistribution, 11
*Topic **misc**
  - assignSeq2REsite, 3
  - buildREmap, 6
  - summarizeByRE, 12
  - summarizeBySeq, 13
*Topic **package**
  - REDseq-package, 2

assignSeq2REsite, 3
binom.test.REDseq, 5
buildREmap, 6
compareREDseq, 7
distanceHistSeq2RE, 8
example.assignedREDseq, 9
example.map, 10
example.REDseq, 10
plotCutDistribution, 11
REDseq (REDseq-package), 2
REDseq-package, 2
summarizeByRE, 12
summarizeBySeq, 13