Package: rnaEditr (via r-universe)

September 29, 2024

Title Statistical analysis of RNA editing sites and hyper-editing regions

Version 1.15.0

Description RNAeditr analyzes site-specific RNA editing events, as well as hyper-editing regions. The editing frequencies can be tested against binary, continuous or survival outcomes. Multiple covariate variables as well as interaction effects can also be incorporated in the statistical models.

Depends R (>= 4.0)

License GPL-3

Encoding UTF-8

LazyData false

RoxygenNote 7.1.1

Imports GenomicRanges, IRanges, BiocGenerics, GenomeInfoDb, bumphunter, S4Vectors, stats, survival, logistf, plyr, corrplot

Suggests knitr, rmarkdown, testthat

biocViews GeneTarget, Epigenetics, DimensionReduction, FeatureExtraction, Regression, Survival, RNASeq

VignetteBuilder knitr

URL https://github.com/TransBioInfoLab/rnaEditr

BugReports https://github.com/TransBioInfoLab/rnaEditr/issues

Repository https://bioc.r-universe.dev

RemoteUrl https://github.com/bioc/rnaEditr

RemoteRef HEAD

RemoteSha 0d74af7d672b6d418a0402a2bfa4917809e6993c

Contents

AllCloseByRegions | *Extract clusters of RNA editing sites located closely in genomic regions.*

Description

A wrapper function to extract clusters of RNA editing sites that are located closely in genomic regions.

Usage

```
AllCloseByRegions(
  regions_gr,
  rnaEditMatrix,
  maxGap = 50,
  minSites = 3,
  progressBar = "time"
)
```

Arguments

regions_gr	A GRanges object of input genomic regions.
rnaEditMatrix	A matrix (or data frame) of RNA editing level values on individual sites, with row names as site IDs in the form of "chrAA:XXXXXXXX", and column names as sample IDs. Please make sure to follow the format of example dataset (data(rnaedit_df)).
maxGap	An integer, genomic locations within maxGap from each other are placed into the same cluster. Defaults to 50.
minSites	An integer, minimum number of RNA editing sites within each resulting cluster. Defaults to 3.
progressBar	Name of the progress bar to use. There are currently five types of progress bars: "time", "none", "text", "tk", and "win". Defaults to "time". See create_progress_bar for more details.

Details

The algorithm of this function is based on the `clusterMaker` function in the bumphunter R package. Each cluster is essentially a group of site locations such that two consecutive locations in the cluster are separated by less than `maxGap`.

Value

A GRanges object containing genomic regions of RNA editing sites located closely within each input pre-defined genomic region.

See Also

`TransformToGR`, `AllCoeditedRegions`, `CreateEditingTable`, `SummarizeAllRegions`, `TestAssociations`, `AnnotateResults`

Examples

```
data(rnaedit_df)

exm_regions <- TransformToGR(
  genes_char = c("PHACTR4", "CCR5", "METTL7A"),
  type = "symbol",
  genome = "hg19"
)

AllCloseByRegions(
  regions_gr = exm_regions,
  rnaEditMatrix = rnaedit_df,
  maxGap = 50,
  minSites = 3,
  progressBar = "time"
)
```

AllCoeditedRegions	*Extracts contiguous co-edited genomic regions from input genomic regions .*

Description

A wrapper function to extract contiguous co-edited genomic regions from input genomic regions.

Usage

```
AllCoeditedRegions(
  regions_gr,
  rnaEditMatrix,
  output = c("GRanges", "dataframe"),
```

```
    rDropThresh_num = 0.4,
    minPairCorr = 0.1,
    minSites = 3,
    method = c("spearman", "pearson"),
    returnAllSites = FALSE,
    progressBar = "time",
    verbose = TRUE
)
```

Arguments

`regions_gr`	A GRanges object of input genomic regions.
`rnaEditMatrix`	A matrix (or data frame) of RNA editing level values on individual sites, with row names as site IDs in the form of "chrAA:XXXXXXXX", and column names as sample IDs. Please make sure to follow the format of example dataset (`data(rnaedit_df)`).
`output`	Type of output data. Defaults to `"GRanges"`.
`rDropThresh_num`	
	Threshold for minimum correlation between RNA editing levels of one site and the mean RNA editing levels of the rest of the sites. Please set a number between 0 and 1. Defaults to 0.4.
`minPairCorr`	Threshold for minimum pairwise correlation of sites within a selected cluster. To use this filter, set a number between -1 and 1 (defaults to 0.1). To select all clusters (i.e. no filter), please set this argument to -1.
`minSites`	Minimum number of sites to be considered as a region. Only regions with more than `minSites` number of sites will be returned.
`method`	Method for computing correlation. Defaults to `"spearman"`.
`returnAllSites`	When no contiguous co-edited regions are found in an input genomic region, `returnAllSites = TRUE` indicates returning all the sites in the input region, while `returnAllSites = FALSE` indicates not returning any site from input region. Defaults to FALSE.
`progressBar`	Name of the progress bar to use. There are currently five types of progress bars: `"time"`, `"none"`, `"text"`, `"tk"`, and `"win"`. Defaults to `"time"`. See [create_progress_bar](#) for more details.
`verbose`	Should messages and warnings be displayed? Defaults to FALSE, but is set to TRUE when called from within `SingleCoeditedRegion()`.

Value

When `output` is set as `"GRanges"`, a GRanges object with `seqnames`, `ranges` and `strand` of the contiguous co-edited regions will be returned. When `output` is set as `"dataframe"`, a data frame with following columns will be returned:

- `site` : site ID.
- `chr` : chromosome number.
- `pos` : genomic position number.

- r_drop : the correlation between RNA editing levels of one site and the mean RNA editing levels of the rest of the sites.
- keep : indicator for co-edited sites, the sites with keep = 1 belong to the contiguous and co-edited region.
- keep_contiguous : contiguous co-edited region number.
- regionMinPairwiseCor : the pairwise correlation of a subregion.
- keep_regionMinPairwiseCor : indicator for contiguous co-edited subregions, the regions with keepminPairwiseCor = 1 passed the minimum correlation and will be returned as a contiguous co-edited subregion.

See Also

[TransformToGR](#), [AllCloseByRegions](#), [CreateEditingTable](#), [SummarizeAllRegions](#), [TestAssociations](#), [AnnotateResults](#)

Examples

```
data(rnaedit_df)

genes_gr <- TransformToGR(
  genes_char = c("PHACTR4", "CCR5", "METTL7A"),
  type = "symbol",
  genome = "hg19"
)

AllCoeditedRegions(
  regions_gr = genes_gr,
  rnaEditMatrix = rnaedit_df,
  output = "GRanges",
  method = "spearman"
)
```

AnnotateResults   *Add Annotations to site-specific or region-based analysis results.*

Description

Add annotations to site-specific or region-based analysis results from function [TestAssociations](#).

Usage

```
AnnotateResults(
  results_df,
  closeByRegions_gr = NULL,
  inputRegions_gr = NULL,
  genome = c("hg38", "hg19"),
  analysis = c("region-based", "site-specific")
)
```

Arguments

`results_df`	An output data frame from function `TestAssociations`, which includes variables for locations and result of statistical tests for the genomic sites or regions.
`closeByRegions_gr`	
	An output GRanges object from function `AllCloseByRegions`, defaults to `NULL`.
`inputRegions_gr`	
	A GRanges object for input genomic regions, defaults to `NULL`.
`genome`	Use `"hg19"` or `"hg38"` gene reference. Defaults to `"hg38"`.
`analysis`	Results type. Defaults to `"region-based"`. When it's set to `"site-specific"`, arguments `closeByRegions_gr` and `inputRegions_gr` will not be used and set to NULL automatically.

Value

A data frame with locations of the genomic sites or regions (`seqnames`, `start`, `end`, `width`), annotations for locations (`inputRegion`, `closeByRegion`, `symbol`), test statistics (`estimate`, `stdErr` or `coef`, `exp_coef`, `se_coef`), pValue and false discovery rate (`fdr`).

See Also

[TransformToGR](#), [AllCloseByRegions](#), [AllCoeditedRegions](#), [CreateEditingTable](#), [SummarizeAllRegions](#), [TestAssociations](#)

Examples

```
data(rnaedit_df)

# get GRanges for genes
genes_gr <- TransformToGR(
  genes_char = c("PHACTR4", "CCR5", "METTL7A"),
  type = "symbol",
  genome = "hg19"
)

# find close-by regions within the genes
closebyRegions_gr <- AllCloseByRegions(
  regions_gr = genes_gr,
  rnaEditMatrix = rnaedit_df
)

# identify co-edited regions within the genes
coedited_gr <- AllCoeditedRegions(
  regions_gr = closebyRegions_gr,
  rnaEditMatrix = rnaedit_df,
  output = "GRanges",
  method = "spearman"
)

# summarize editing levels within each gene by maximum
summarizedRegions_df <- SummarizeAllRegions(
```

```
    regions_gr = coedited_gr,
    rnaEditMatrix = rnaedit_df,
    selectMethod = MaxSites
  )

  exm_pheno <- readRDS(
    system.file(
    "extdata",
    "pheno_df.RDS",
    package = 'rnaEditr',
    mustWork = TRUE
    )
  )

  # test summarized editing levels against survival outcome
  results_df <- TestAssociations(
    rnaEdit_df = summarizedRegions_df,
    pheno_df = exm_pheno,
    responses_char = "sample_type",
    covariates_char = NULL,
    respType = "binary"
  )

  AnnotateResults(
    results_df = results_df,
    closeByRegions_gr = closebyRegions_gr,
    inputRegions_gr = genes_gr,
    genome = "hg19"
  )
```

CreateEditingTable   *Convert RNA editing matrix into a special data frame with class*
                            rnaEdit_df.

Description

Convert RNA editing matrix to a special data frame with class rnaEdit_df, which is then used to
identify differentially co-edited regions with function `TestAssociations`.

Usage

```
CreateEditingTable(rnaEditMatrix)
```

Arguments

rnaEditMatrix   A matrix of RNA editing level values on individual sites, with row names as site
                IDs in the form of "chrAA:XXXXXXXX", and column names as sample IDs.
                Please make sure to follow the format of example dataset (data(rnaedit_df)).

Value

A dataset of class rnaEdit_df, includes variables seqnames, start, end, width and summarized RNA editing levels in each sample.

See Also

[TransformToGR](), [AllCloseByRegions](), [AllCoeditedRegions](), [SummarizeAllRegions](), [TestAssociations](),
[AnnotateResults]()

Examples

```
data(rnaedit_df)
CreateEditingTable(rnaEditMatrix = rnaedit_df)[1:3, 1:5]
```

---

   rnaedit_df   *Example breast cancer RNA editing dataset.*

Description

A subset of the TCGA breast cancer RNA editing dataset for 272 edited sites on genes PHACTR4, CCR5, METTL7A and a few randomly sampled sites for 221 subjects.

Usage

```
rnaedit_df
```

Format

A data frame containing RNA editing levels for 272 sites (in the rows) for 221 subjects (in the columns). Row names are site IDs and column names are sample IDs.

Source

Synapse database ID: syn2374375.

SummarizeAllRegions   *Summarize RNA editing levels from multiple sites in regions.*

Description

A wrapper function to summarize RNA editing levels from multiple sites in regions.

Usage

```
SummarizeAllRegions(
  regions_gr,
  rnaEditMatrix,
  selectMethod = MedianSites,
  progressBar = "time",
  ...
)
```

Arguments

regions_gr	A GRanges object of input genomic regions.
rnaEditMatrix	A matrix (or data frame) of RNA editing level values for individual sites, with row names as site IDs in the form of "chrAA:XXXXXXXX", and column names as sample IDs. Please make sure to follow the format of example dataset (data(rnaedit_df)).
selectMethod	Method for summarizing regions. Available options are "MaxSites", "MeanSites", "MedianSites", "PC1Sites". Please see [RegionSummaryMethod](#) for more details.
progressBar	Name of the progress bar to use. There are currently five types of progress bars: "time", "none", "text", "tk", and "win". Defaults to "time". See [create_progress_bar](#) for more details.
...	Dots for additional internal arguments (currently unused).

Value

A data frame of the class rnaEdit_df, includes variables seqnames, start, end, width and summarized RNA editing levels in each sample.

See Also

[TransformToGR](#), [AllCloseByRegions](#), [AllCoeditedRegions](#), [CreateEditingTable](#), [TestAssociations](#), [AnnotateResults](#)

Examples

```
data(rnaedit_df)

genes_gr <- TransformToGR(
  genes_char = c("PHACTR4", "CCR5", "METTL7A"),
```

```
    type = "symbol",
    genome = "hg19"
  )

  exm_regions <- AllCoeditedRegions(
    regions_gr = genes_gr,
    rnaEditMatrix = rnaedit_df,
    output = "GRanges",
    method = "spearman"
  )

  SummarizeAllRegions(
    regions_gr = exm_regions,
    rnaEditMatrix = rnaedit_df
  )[1:3, 1:6]
```

TestAssociations   *Test associations between phenotype and RNA editing levels.*

Description

A wrapper function to test associations between phenotype and RNA editing levels in single-site analysis or summarized RNA editing levels in region-based analysis.

Usage

```
TestAssociations(
  rnaEdit_df,
  pheno_df,
  responses_char,
  covariates_char = NULL,
  respType = c("binary", "continuous", "survival"),
  progressBar = "time",
  orderByPval = TRUE
)
```

Arguments

rnaEdit_df   A data frame with class rnaEdit_df, which is a output from function [CreateEditingTable]()
                or function [SummarizeAllRegions](). This data frame should include RNA
                editing level values, with row names as site IDs or region IDs, and column names
                as sample IDs.

pheno_df   A data frame with phenotype and covariates, which should include all the sam-
                ples in rnaEdit_df. Please make sure the input pheno_df has the variable
                named "sample" to indicate sample IDs.

responses_char   A character vector of names of response variables in pheno_df. When respType
is set as `"survival"`, responses_char should have length 2. The first element
must be the name of the variable with following up time, and the second ele-
ment must be status indicator. Status indicator should be coded as 0/1(1=death),
TRUE/FALSE(TRUE=death), or 1/2(death). Please make sure variable names
are in this order. We have not tested this code on interval-censored data; use at
your own risk. See [Surv](#) for more details.

covariates_char

A character vector of names of covariate variables in pheno_df.

respType   Type of outcome. Defaults to `"binary"`.

progressBar   Name of the progress bar to use. There are currently five types of progress
bars: `"time"`, `"none"`, `"text"`, `"tk"`, and `"win"`. Defaults to `"time"`. See
[create_progress_bar](#) for more details.

orderByPval   Sort co-edited regions by model p-value or not? Defaults to TRUE.

Value

A data frame with locations of the genomic regions or sites (`seqnames`, `start`, `end`, `width`),
test statistics (`estimate`, `stdErr` or `coef`, `exp_coef`, `se_coef`), `pValue` and false discovery rate
(`fdr`).

See Also

[TransformToGR](#), [AllCloseByRegions](#), [AllCoeditedRegions](#), [CreateEditingTable](#), [SummarizeAllRegions](#),
[AnnotateResults](#)

Examples

```
data(rnaedit_df)

genes_gr <- TransformToGR(
  genes_char = c("PHACTR4", "CCR5", "METTL7A"),
  type = "symbol",
  genome = "hg19"
)

exm_regions <- AllCoeditedRegions(
  regions_gr = genes_gr,
  rnaEditMatrix = rnaedit_df,
  output = "GRanges",
  method = "spearman"
)

sum_regions <- SummarizeAllRegions(
  regions_gr = exm_regions,
  rnaEditMatrix = rnaedit_df,
  selectMethod = MaxSites
)

exm_pheno <- readRDS(
```

```
    system.file(
    "extdata",
    "pheno_df.RDS",
    package = 'rnaEditr',
    mustWork = TRUE
    )
  )

  TestAssociations(
    rnaEdit_df = sum_regions,
    pheno_df = exm_pheno,
    responses_char = "sample_type",
    covariates_char = NULL,
    respType = "binary"
  )
```

TransformToGR   *Transform gene symbols or region ranges into GRanges object.*

Description

Transform a character vector of gene symbols or region ranges into a GRanges object.

Usage

```
TransformToGR(
  genes_char,
  type = c("symbol", "region"),
  genome = c("hg38", "hg19")
)
```

Arguments

genes_char   A character vector of gene symbols or region ranges. If you select type to be
                "symbol", then please make sure your input of genes_char is in the format of
                c("ABCB10", "PEX26"). If you select type to be "region", then please make
                sure your input of genes_char is in the format of c("chr1:33772367-33791699",
                "chr22:18555686-18573797").

type   What is the type of genes_char. Can be "symbol" (default) or "region".

genome   Use "hg19" or "hg38" gene reference. Defaults to "hg38". It's only used when
                type is set to "symbol"

Details

TransformToGR() uses the hg19/hg38 genes to associate gene symbols with their genomic region
ranges. The pre-processed dataset is saved in inst/extdata in this package.

Users who wish to add gene symbols to the GRanges created using function TransformToGR() can
use function AddMetaData(). Please see [AddMetaData](#) for details.

Value

A GRanges object with seqnames, ranges and strand.

See Also

[AllCloseByRegions](), [AllCoeditedRegions](), [CreateEditingTable](), [SummarizeAllRegions](), [TestAssociations](), [AnnotateResults]()

Examples

```
TransformToGR(
  genes_char = c("PHACTR4", "CCR5", "METTL7A"),
  type = "symbol",
  genome = "hg19"
)

TransformToGR(
  genes_char = c("chr22:18555686-18573797", "chr22:36883233-36908148"),
  type = "region",
  genome = "hg19"
)
```

---

t_rnaedit_df   *Transposed breast cancer example dataset.*

Description

A subset of the TCGA breast cancer RNA editing dataset for 20 randomly selected RNA editing sites and 50 randomly selected subjects from example dataset rnaedit_df. Please note that this is only a computational testing dataset for inner functions of this package. To test main functions, please use dataset rnaedit_df instead.

Usage

```
t_rnaedit_df
```

Format

A data frame containing RNA editing levels for 50 subjects (in the rows) at 20 edited sites (in the columns). Row names are sample IDs and column names are site IDs.

Source

Synapse database ID: syn2374375.

Index