Package: Site2Target (via r-universe)

May 31, 2026

Type Package

Title An R package to associate peaks and target genes

Version 1.4.0

Description Statistics implemented for both peak-wise and gene-wise associations. In peak-wise associations, the p-value of the target genes of a given set of peaks are calculated. Negative binomial or Poisson distributions can be used for modeling the unweighted peaks targets and log-nromal can be used to model the weighted peaks. In gene-wise associations a table consisting of a set of genes, mapped to specific peaks, is generated using the given rules.

BugReports https://github.com/fls-bioinformatics-core/Site2Target/issues

Depends R (>= 4.4)

License GPL-2

Encoding UTF-8

LazyData false

Imports S4Vectors, stats, utils, BiocGenerics, GenomeInfoDb, MASS, IRanges, GenomicRanges

biocViews Annotation, ChIPSeq, Software, Epigenetics, GeneExpression, GeneTarget

RoxygenNote 7.3.2

Suggests BiocStyle, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/pak/sysreqs libssl-dev

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

Date/Publication 2026-04-28 13:04:44 UTC

RemoteUrl https://github.com/bioc/Site2Target

RemoteRef RELEASE_3_23

RemoteSha ce9001280a619e0fc8dacf0ca1bd037c984e5d08

Contents

`addColumn2geneWiseAssociation`

Add column to gene-wise association

Description

Add a column of values based on the type either genes or peaks.

The Input is either coordinates or names of genes or peaks plus

a column of relevant values. This function add these values as

a column to gene or peak table as well as the interaction table.

Usage

```
addColumn2geneWiseAssociation(
  type = "",
  name = NULL,
  coordinates = NULL,
  columnName = NA,
  column,
  inFile = "geneWiseAssociation",
  outFile = "geneWiseAssociation"
)
```

Arguments

type	type of columns to be added. Either "gene" or "peak"
name	Names of genes or peaks
coordinates	Coordinates of genes or peaks in granges format
columnName	Column name that should be added to the tables
column	Column values that should be added to the tables
inFile	The name of the input folder (default "genewiseAssociation")
outFile	The name of the output folder (default "genewiseAssociation")

Value

No value returns just column would be added to the tables

See Also

genewiseAssociation

Examples

```
geneFile=system.file("extdata", "gene_expression.tsv", package="Site2Target")
geneCoords <- Table2Granges(geneFile)
geneTable <- read.table(geneFile, header=TRUE)

geneDEIndices <- which((abs(geneTable$logFC)>1)==TRUE)
indicesLen <- length(geneDEIndices)
if(indicesLen >0)
{
    geneTable <- geneTable[geneDEIndices,]
    geneCoords <- geneCoords[geneDEIndices]
}
geneDENames <- geneTable$name
geneDElogFC <- geneTable$logFC
geneCoordsDE <- geneCoords

tfFile =system.file("extdata", "MEIS_binding.tsv", package="Site2Target")
TFCoords <- Table2Granges(tfFile)
tfTable <- read.table(tfFile, header=TRUE)
tfIntensities <- tfTable$intensities

stats <-
genewiseAssociation(associationBy="distance",
                    geneCoordinates=geneCoordsDE,
                    geneNames=geneDENames,
                    peakCoordinates=TFCoords,
                    distance=50000,
                    outFile="Gene_TF_50K")
stats

# add expression log fold changes to the table
```

```
addColumn2geneWiseAssociation(type="gene", name=geneDENames,
   columnName="Expr_logFC", column=geneDElogFC, inFile="Gene_TF_50K",
   outFile="Gene_TF_50K")

# add peak intensitites to the table
addColumn2geneWiseAssociation(type="peak", coordinates=TFCoords,
   columnName="Binding_Intensities", column=tfIntensities,
   inFile="Gene_TF_50K", outFile="Gene_TF_50K")
```

`addRelation2geneWiseAssociation` *Add a relation column to gene-peak interaction table*

Description

Get coordinates of interactions (ex. HiC interactions) and a column of interaction values (ex. HiC intensities) and add them as a column to gene-peak interaction table.

Usage

```
addRelation2geneWiseAssociation(
  strand1 = NULL,
  strand2 = NULL,
  columnName,
  column,
  inFile = "geneWiseAssociation",
  outFile = "geneWiseAssociation"
)
```

Arguments

`strand1`	granges of DNA strand1 linked to DNA strand2
`strand2`	granges of DNA strand2 linked to DNA strand1
`columnName`	Column name that should be added to the interaction table
`column`	Column values that should be added to the interaction table
`inFile`	The name of the input folder (default "genewiseAssociation")
`outFile`	The name of the output folder (default "genewiseAssociation")

Value

No value would be returned just a column be added to link table

See Also

`genewiseAssociation`

Examples

```
geneFile=system.file("extdata", "gene_expression.tsv", package="Site2Target")
geneCoords <- Table2Granges(geneFile)
geneTable <- read.table(geneFile, header=TRUE)

geneDEIndices <- which((abs(geneTable$logFC)>1)==TRUE)
indicesLen <- length(geneDEIndices)
if(indicesLen >0)
{
    geneTable <- geneTable[geneDEIndices,]
    geneCoords <- geneCoords[geneDEIndices]
}
geneDENames <- geneTable$name
geneDElogFC <- geneTable$logFC
geneCoordsDE <- geneCoords

tfFile =system.file("extdata", "MEIS_binding.tsv", package="Site2Target")
TFCoords <- Table2Granges(tfFile)
tfTable <- read.table(tfFile, header=TRUE)

stats <-
genewiseAssociation(associationBy="distance",
                    geneCoordinates=geneCoordsDE,
                    geneNames=geneDENames,
                    peakCoordinates=TFCoords,
                    distance=50000,
                    outFile="Gene_TF_50K")
stats

HiCFile =system.file("extdata", "HiC_intensities.tsv", package="Site2Target")
HiCstr1 <- Table2Granges(HiCFile, chrColName="Strand1_chr",
                     startColName="Strand1_start", endColName="Strand1_end")
HiCstr2 <- Table2Granges(HiCFile, chrColName="Strand2_chr",
                     startColName="Strand2_start", endColName="Strand2_end")
HiCTable <- read.table(HiCFile, header=TRUE)
HiCintensities <- HiCTable$intensities

addRelation2geneWiseAssociation(strand1=HiCstr1, strand2=HiCstr2,
     columnName="HiC_Intensities", column=HiCintensities,
     inFile="Gene_TF_50K", outFile="Gene_TF_50K")
```

`data` *MEIS cardiomyocytes datasets used in the package*

Description

Human cardiomyocytes datasets are reduced in size by only using chr21. log fold changes of Gene expression WT vs MEIS KO from RNA-seq experiments, and binding sites of MEIS derived from a ChIP-seq experiment are the main experimental datasets representing relevant gene and peak information. HiC interactions and topologically associating domains (TADs) are derived from a HiC experiments are auxiliary datasets related to DNA-DNA interactions.

Format

Gene expression WT vs MEIS KO in chr21. MEIS binding sites in chr21. TADs, and HiC interactions in chr21.

gene_expression.tsv Gene expression

MEIS_binding.tsv MEIS binding sites

TADs.tsv TADs

HiC_intensities.tsv HiC interactions

Value

Just description of data

Examples

```
## Gene expression table

# Read gene coordinates
geneFile=system.file("extdata", "gene_expression.tsv",
                     package="Site2Target")
geneCoords <- Table2Granges(geneFile)

# Read gene table
geneTable <- read.table(geneFile, header=TRUE)



## TF binding table

# Read peak coordinates
tfFile =system.file("extdata", "MEIS_binding.tsv",
                    package="Site2Target")
TFCoords <- Table2Granges(tfFile)

# Read MEIS binding intensities
tfTable <- read.table(tfFile, header=TRUE)


## DNA-DNA interactions

# Read TAD regions
TADsFile =system.file("extdata", "TADs.tsv",
                              package="Site2Target")
TADs <- Table2Granges(TADsFile)


# Read HiC interactions
HiCFile =system.file("extdata", "HiC_intensities.tsv",
                              package="Site2Target")
HiCstr1 <- Table2Granges(HiCFile, chrColName="Strand1_chr",
                      startColName="Strand1_start", endColName="Strand1_end")
```

```
HiCstr2 <- Table2Granges(HiCFile, chrColName="Strand2_chr",
                     startColName="Strand2_start", endColName="Strand2_end")

HiCTable <- read.table(HiCFile, header=TRUE)
```

`extendSitesInGivenRegions` *Extend sites given regions boundaries*

Description

Get sites and given regions (ex. TADs or loops) coordinates.

It extends sites in a give region using a distance function

Usage

```
extendSitesInGivenRegions(givenRegions, sites, distance = 1e+05)
```

Arguments

`givenRegions`	granges coordinates of given regions (ex. TAD or loops)
`sites`	granges coordinates of sites
`distance`	the maximum distance to associate sites to regions

Value

A granges of the extended sites in given regions

Examples

```
tfFile =system.file("extdata", "MEIS_binding.tsv", package="Site2Target")
TFCoords <- Table2Granges(tfFile)

TADsFile =system.file("extdata", "TADs.tsv",package="Site2Target")
TADs <- Table2Granges(TADsFile)

extendSitesInGivenRegions(TADs, TFCoords)
```

genewiseAssociation *Generate genewise association between genes and peaks*

Description

Get genomic coordinates of a set of genes and a set of peaks associate them by a fixed distance (default 50K nt). It also associate genes and peaks for provided DNA-DNA interaction from a dataset like HiC. This function can also associate genes and user provided regions (ex. TADs, subTADs, etc). It generates three tables: Gene table, peak table, and Gene-Peak association table.

Usage

```
genewiseAssociation(
  associationBy = "distance",
  geneCoordinates = NULL,
  geneNames = NULL,
  peakCoordinates = NULL,
  peakNames = NULL,
  distance = 50000,
  givenRegions = NULL,
  strand1 = NULL,
  strand2 = NULL,
  outFile = "genewiseAssociation"
)
```

Arguments

associationBy	Can be "distance", "regions", or "DNAinteractions"
geneCoordinates	Gene coordinates in granges format
geneNames	Gene names can be provided by the user
peakCoordinates	Peak coordinates in granges format
peakNames	Peak names can be provided by the user
distance	The maximum distance to associate peaks to genes. default 50K
givenRegions	granges coordinates of given regions (ex. TAD or loops)
strand1	granges of DNA strand1 linked to DNA strand2
strand2	granges of DNA strand2 linked to DNA strand1
outFile	The name of the output folder (default "genewiseAssociation")

Value

A vector of portions of linked genes and linked peaks

Examples

```
geneFile=system.file("extdata", "gene_expression.tsv", package="Site2Target")
geneCoords <- Table2Granges(geneFile)
geneTable <- read.table(geneFile, header=TRUE)

geneDEIndices <- which((abs(geneTable$logFC)>1)==TRUE)
indicesLen <- length(geneDEIndices)
if(indicesLen >0)
{
    geneTable <- geneTable[geneDEIndices,]
    geneCoords <- geneCoords[geneDEIndices]
}
geneDENames <- geneTable$name
geneDElogFC <- geneTable$logFC
geneCoordsDE <- geneCoords

tfFile =system.file("extdata", "MEIS_binding.tsv", package="Site2Target")
TFCoords <- Table2Granges(tfFile)
tfTable <- read.table(tfFile, header=TRUE)

stats <-
genewiseAssociation(associationBy="distance",
                    geneCoordinates=geneCoordsDE,
                    geneNames=geneDENames,
                    peakCoordinates=TFCoords,
                    distance=50000,
                    outFile="Gene_TF_50K")
stats
```

---

getCenterOfPeaks *Return center of the given granges files*

Description

Get a granges and find the center of it

Usage

```
getCenterOfPeaks(gr)
```

Arguments

gr	granges coordinate

Value

granges format of the center

Examples

```
tfFile =system.file("extdata", "MEIS_binding.tsv", package="Site2Target")
TFCoords <- Table2Granges(tfFile)
TFCoordsCenters <- getCenterOfPeaks(TFCoords)
TFCoordsCenters
```

---

getNameFromCoordinates

Get names of genes or peaks related to a query coordinates

Description

Get names and coordinates of genes or peaks. It also get the coordinates of query regions and returns the related genes or peak names.

Usage

```
getNameFromCoordinates(names, coordinates, queryCoordinates)
```

Arguments

names	Names of genes or peaks
coordinates	Coordinates of genes or peaks in granges format
queryCoordinates	Coordinates of the query regions in granges format

Value

Names of genes or peaks in queried regions

getTargetGenesNumber *generate number of sites per gene given distances*

Description

Get genes and sites coordinates, and associate them by given distance.

Usage

```
getTargetGenesNumber(geneCoordinates = NA, sites = NA, distance = 50000)
```

Arguments

geneCoordinates	granges coordinates of genes
sites	granges coordinates of sites
distance	the maximum distance to associate sites to genes. default 50K

Value

A vector sites number matched to each gene

Examples

```
geneFile=system.file("extdata", "gene_expression.tsv", package="Site2Target")
geneCoords <- Table2Granges(geneFile)

tfFile =system.file("extdata", "MEIS_binding.tsv", package="Site2Target")
TFCoords <- Table2Granges(tfFile)

targetNum <- getTargetGenesNumber( geneCoords, TFCoords)
```

getTargetGenesPvals *Fit Negative binomial distribution to target genes*

Description

Get genes and sites coordinates, and associate them by given distance or given regions (ex. TADs or loops). It tests the distribution of sites around genes either by poisson or negative binomial test.

Usage

```
getTargetGenesPvals(
  associationBy = "distance",
  dist = "negative binomial",
  geneCoordinates = NA,
  sites = NA,
  distance = 50000,
  givenRegions = NA
)
```

Arguments

associationBy	either "distance" or "regions"
dist	either "negative binomial" or "poisson"
geneCoordinates	granges coordinates of genes
sites	granges coordinates of sites
distance	the maximum distance to associate sites to genes. default 50K
givenRegions	user provided granges regions like TADs or loops

Value

A vector of pvalue distribution for target genes

Examples

```
geneFile=system.file("extdata", "gene_expression.tsv", package="Site2Target")
geneCoords <- Table2Granges(geneFile)

tfFile =system.file("extdata", "MEIS_binding.tsv", package="Site2Target")
TFCoords <- Table2Granges(tfFile)

pvals <- getTargetGenesPvals( geneCoordinates=geneCoords, sites=TFCoords)
```

---

getTargetGenesPvalsWithDNAInteractions

Fit Negative binomial distribution to target genes

Description

Get genes and sites coordinates, and associate them by given distance and user provided DNA interaction (ex. HiC). It tests the distribution of sites around genes either by poisson or negative binomial test.

Usage

```
getTargetGenesPvalsWithDNAInteractions(
  dist = "negative binomial",
  geneCoordinates = NA,
  sites = NA,
  strand1 = NA,
  strand2 = NA,
  distance = 50000
)
```

Arguments

dist	either "negative binomial" or "poisson"
geneCoordinates	granges coordinates of genes
sites	granges coordinates of sites
strand1	granges of DNA strand1 linked to DNA strand2
strand2	granges of DNA strand2 linked to DNA strand1
distance	the maximum distance to associate sites to genes. default 50K

Value

A vector of pvalue distribution for target genes

Examples

```
geneFile=system.file("extdata", "gene_expression.tsv", package="Site2Target")
geneCoords <- Table2Granges(geneFile)

tfFile =system.file("extdata", "MEIS_binding.tsv", package="Site2Target")
TFCoords <- Table2Granges(tfFile)

HiCFile =system.file("extdata", "HiC_intensities.tsv", package="Site2Target")
HiCstr1 <- Table2Granges(HiCFile, chrColName="Strand1_chr",
                    startColName="Strand1_start", endColName="Strand1_end")
HiCstr2 <- Table2Granges(HiCFile, chrColName="Strand2_chr",
                    startColName="Strand2_start", endColName="Strand2_end")

pvals <- getTargetGenesPvalsWithDNAInteractions(
               geneCoordinates=geneCoords, sites=TFCoords, strand1=HiCstr1,
               strand2=HiCstr2)
```

`getTargetGenesPvalsWithIntensities` *Fit log-normal distribution to target genes*

Description

Get genes and sites coordinates, and associate them by given distance or given regions (ex. TADs or loops). It tests the distribution of log-intensities of sites around genes by log-normal test. This function consider both binding sites and intensities.

Usage

```
getTargetGenesPvalsWithIntensities(
  associationBy = "distance",
  intensities,
  geneCoordinates = NA,
  sites = NA,
  distance = 50000,
  givenRegions = NA
)
```

Arguments

`associationBy`	either "distance" or "regions"
`intensities`	intensity values associated to sites
`geneCoordinates`	granges coordinates of genes
`sites`	granges coordinates of sites
`distance`	the maximum distance to associate sites to genes. default 50K
`givenRegions`	user provided granges regions like TADs or loops

Value

A vector of pvalue distribution for target genes

Examples

```
geneFile=system.file("extdata", "gene_expression.tsv", package="Site2Target")
geneCoords <- Table2Granges(geneFile)

tfFile =system.file("extdata", "MEIS_binding.tsv", package="Site2Target")
TFCoords <- Table2Granges(tfFile)
tfTable <- read.table(tfFile, header=TRUE)
```

```
tfIntensities <- tfTable$intensities

pvals <- getTargetGenesPvalsWithIntensities(geneCoordinates=geneCoords,
                      sites=TFCoords, intensities=tfIntensities)
```

`getTargetGenesPvalsWithIntensitiesAndDNAInteractions`
Fit log-normal distribution to target genes

Description

Get genes and sites coordinates, and associate them by given distance and user provided DNA interaction (ex. HiC). It tests the distribution of log-intensities of sites around genes by log-normal test. This function consider both binding sites and intensities.

Usage

```
getTargetGenesPvalsWithIntensitiesAndDNAInteractions(
  geneCoordinates,
  sites,
  intensities,
  strand1,
  strand2,
  distance = 50000
)
```

Arguments

`geneCoordinates`	granges coordinates of genes
`sites`	granges coordinates of sites
`intensities`	intensity values associated to sites
`strand1`	granges of DNA strand1 linked to DNA strand2
`strand2`	granges of DNA strand2 linked to DNA strand1
`distance`	the maximum distance to associate sites to genes. default 50K

Value

A vector of pvalue distribution for target genes

Examples

```
geneFile=system.file("extdata", "gene_expression.tsv", package="Site2Target")
geneCoords <- Table2Granges(geneFile)

tfFile =system.file("extdata", "MEIS_binding.tsv", package="Site2Target")
TFCoords <- Table2Granges(tfFile)
tfTable <- read.table(tfFile, header=TRUE)
tfIntensities <- tfTable$intensities

HiCFile =system.file("extdata", "HiC_intensities.tsv", package="Site2Target")
HiCstr1 <- Table2Granges(HiCFile, chrColName="Strand1_chr",
                     startColName="Strand1_start", endColName="Strand1_end")
HiCstr2 <- Table2Granges(HiCFile, chrColName="Strand2_chr",
                     startColName="Strand2_start", endColName="Strand2_end")

pvals <- getTargetGenesPvalsWithIntensitiesAndDNAInteractions(
                       geneCoordinates=geneCoords, sites=TFCoords,
                       intensities=tfIntensities, strand1=HiCstr1,
                        strand2=HiCstr2)
```

`granges2String` *Convert granges to strings of coordinates*

Description

Get genomic coordinates granges and convert them to strings

Usage

```
granges2String(gr)
```

Arguments

`gr`	granges coordinates

Value

string of coordinates

Examples

```
tfFile =system.file("extdata", "MEIS_binding.tsv", package="Site2Target")
TFCoords <- Table2Granges(tfFile)
strCoords <- granges2String(TFCoords)
head(strCoords)
```

removeReserveCharacter *Remove reserved characters from a string*

Description

Remove reserved characters (such as *, +, -, etc) from a string

Usage

```
removeReserveCharacter(name)
```

Arguments

name	A string of characters

Value

A string without reserved characters

Examples

```
removeReserveCharacter("A&%B^f6")
```

site2GeneDistance *Return the distance between paired peaks and genes*

Description

Get a granges of genes and peaks and return their distances

Usage

```
site2GeneDistance(geneCoordinates, peakCoordinates)
```

Arguments

geneCoordinates	granges coordinates of genes
peakCoordinates	granges coordinates of peaks

Value

the respective distances of paired genes and peaks

Site2Target *Associate peaks and target genes*

Description

Statistical implementation for both peak-wise and gene-wise associations. Here is an example of a peak-wise and a gene-wise association of differential genes WT vs KO of a transcription factor and binding sites of this transcription factor.

Value

Just an example

Examples

```
geneFile=system.file("extdata", "gene_expression.tsv", package="Site2Target")
geneCoords <- Table2Granges(geneFile)
geneTable <- read.table(geneFile, header=TRUE)

tfFile =system.file("extdata", "MEIS_binding.tsv", package="Site2Target")
TFCoords <- Table2Granges(tfFile)
tfTable <- read.table(tfFile, header=TRUE)

## Peakwise association example

pvals <- getTargetGenesPvals(geneCoordinates=geneCoords, sites=TFCoords)
topTargetNum <- 5
topTargetIndex <- order(pvals)[1:topTargetNum]

# Make a data frame of peak targets pvalues and expression logFCs

dfTopTarget <-
  data.frame(name=geneTable$name[topTargetIndex],
             pvalue=pvals[topTargetIndex],
             exprLogC=geneTable$logFC[topTargetIndex]
             )
dfTopTarget

## Genewise association example
geneDEIndices <- which((abs(geneTable$logFC)>1)==TRUE)
indicesLen <- length(geneDEIndices)
if(indicesLen >0)
{
    geneTable <- geneTable[geneDEIndices,]
    geneCoords <- geneCoords[geneDEIndices]
}
geneDENames <- geneTable$name
geneDElogFC <- geneTable$logFC
geneCoordsDE <- geneCoords
```

```
stats <-
genewiseAssociation(associationBy="distance",
                    geneCoordinates=geneCoordsDE,
                    geneNames=geneDENames,
                    peakCoordinates=TFCoords,
                    distance=50000,
                    outFile="Gene_TF_50K")
stats
```

`string2Granges` *Convert strings to granges of coordinates*

Description

Get genomic coordinates as strings and convert them to granges

Usage

```
string2Granges(strCoordinates)
```

Arguments

`strCoordinates`	string of coordinates

Value

Genomic coordinates in granges format

Examples

```
string2Granges(c("chr1:1112-1231", "ch2:3131-3221"))
```

`Table2Granges` *Take Genomic Ranges from a table file*

Description

Read a table file and derive genomic ranges from user provided column names.

Usage

```
Table2Granges(
  fileName,
  chrColName = "chr",
  startColName = "start",
  endColName = "end"
)
```

Arguments

fileName	A table delimited file
chrColName	Chromosomes column name (default: "Chr")
startColName	Start column name (default: "start")
endColName	End column name (default: "end")

Value

granges format of given coordinates

Examples

```
geneFile=system.file("extdata", "gene_expression.tsv", package="Site2Target")
grs <- Table2Granges(fileName=geneFile,
                     chrColName="chr",
                      startColName="start",
                      endColName="end")
grs
```

Index