Package ‘HiCociety’

May 13, 2025

Type Package

Title Inferring Chromatin Interaction Modules from 3C-Based Data

Version 0.1.38

Author Sora Yoon [aut, cre]

Maintainer Sora Yoon <`sora.yoon@pennmedicine.upenn.edu`>

Description Identifies chromatin interaction modules by constructing a Hi-C contact network based on statistically significant interactions, followed by network clustering. The method enables comparison of module connectivity across two Hi-C datasets and is capable of detecting cell-type-specific regulatory modules. By integrating network analysis with chromatin conformation data, this approach provides insights into the spatial organization of the genome and its functional implications in gene regulation. Author: Sora Yoon (2025) <`https://github.com/ysora/HiCociety`>.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 3.5.0)

Imports strawr, shape, fitdistrplus, igraph, ggraph, foreach, doParallel, biomaRt, TxDb.Hsapiens.UCSC.hg38.knownGene, TxDb.Mmusculus.UCSC.mm10.knownGene, org.Mm.eg.db, org.Hs.eg.db, Rcpp, AnnotationDbi, GenomicFeatures, parallel, IRanges, S4Vectors, grDevices, graphics, stats, BiocManager, BiocGenerics, GenomicRanges, pracma, signal, HiCocietyExample

LinkingTo Rcpp

NeedsCompilation yes

Repository CRAN

Date/Publication 2025-05-13 08:20:02 UTC

Contents

`add_Genes` *Add gene information*

Description

This function adds a column with a list of genes included in each locus to the ModuleSummary data frame of the hic2community function.

Usage

```
add_Genes(df, speciesObj)
```

Arguments

`df`	The ModuleSummary data frame obtained by running hic2community function
`speciesObj`	Any Txdb package name corresponding

Details

Adding gene list to ModuleSummary data frame obtained from hic2community function.

Value

A `data.frame` identical to the input, with an additional `"Genes"` column. Each entry in this column lists the gene(s) that overlap with the corresponding genomic region. If multiple genes are present, they are concatenated with commas.

Author(s)

Sora Yoon, PhD

Examples

```
modulefile = system.file('extdata','mouse_naiveCD4T_Vahedi_short.rds',
package = 'HiCocietyExample')
mycom = readRDS(modulefile)
mycom$ModuleSummary = add_Genes(mycom$ModuleSummary,
'TxDb.Mmusculus.UCSC.mm10.knownGene')
```

`calculate_avg_count` *Calculate Average Count within 5-pixel Padding*

Description

This function calculates the average count within a 25kb padding around each (x, y) coordinate pair.

Usage

```
calculate_avg_count(x, y, counts, resol)
```

Arguments

`x`	Numeric vector of x-coordinates of contact frequency data frame.
`y`	Numeric vector of y-coordinates of contact frequency data frame.
`counts`	Numeric vector of contact frequency counts.
`resol`	Integer specifying the HiC resolution.

Value

A numeric vector of average counts.

Examples

```
x <- c(1, 2, 3, 4, 5)
y <- c(1, 2, 3, 4, 5)
counts <- c(10, 20, 30, 40, 50)
resol <- 10000
calculate_avg_count(x, y, counts, resol)
```

`check_package` *Check if a package is installed*

Description

This function checks whether a package is installed. If the package is not installed, it informs the user that the package is missing.

Usage

```
check_package(package)
```

Arguments

`package`	The name of the package to check.

Details

This function checks whether a package is installed. If the package is not installed, it informs the user that the package is missing.

Value

A logical value: TRUE if the package is installed, FALSE otherwise.

Author(s)

Sora Yoon, PhD

Examples

```
check_package('dplyr')
```

`ConnectivityDiff` *Connectivity difference between two conditions*

Description

output table of connectivity difference of modules between cell types is generated.

Usage

```
ConnectivityDiff(wt, ko, prefix.wt, prefix.ko, resolution = 5000)
```

Arguments

`wt`	hic2community result from condition 1
`ko`	hic2community result from condition 2
`prefix.wt`	Prefix for wt to be presented in the column names
`prefix.ko`	Prefix for ko to be presented in the column names
`resolution`	Resolution of Hi-C dataset

Details

Connectivity difference between two conditions

Value

A `list` of two `data.frame` objects, each representing the network connectivity differences of modules in condition 1 or condition 2 when compared to the counterpart cell type. Each `data.frame` contains the following columns: `"chr"`, `"module_start"`, `"module_end"`, `"connectivity"`, `"transitivity"`, `"centrality_node"`, `"idx"` (the row index of the module in the input module object), `"connectivity_in_(counterpart_c`
`"connectivity_difference"`, and `"connectivity_foldchange"`.

Author(s)

Sora Yoon, PhD

Examples

```
modulefile1 = system.file('extdata','mouse_naiveCD4T_Vahedi_short.rds',
package = 'HiCocietyExample')
modulefile2 = system.file('extdata','mouse_Th1_Vahedi_short.rds',
package = 'HiCocietyExample')
mycom1 = readRDS(modulefile1)
mycom2 = readRDS(modulefile2)
result = ConnectivityDiff(mycom1, mycom2, 'NaiveCD4T', 'Th1',
resolution = 5000)
head(print(result))
```

`getContactFrequency` *Get contact Frequency*

Description

Retrieve contact frequency from .hic file using strawR package.

Usage

```
getContactFrequency(fname, chr, resol)
```

Arguments

`fname`	.hic data for any types of genome conformation capture data
`chr`	Chromosome number of network extraction
`resol`	DNA basepair Resolution. Default=10000

Details

Get Contact Frequncy from .hic file

Value

A `data.frame` containing three columns: `x` and `y` (genomic coordinate pairs), and `counts` (the contact frequency between them).

Author(s)

Sora Yoon, PhD

Examples

```
myhic=system.file('extdata', 'example.hic', package ='HiCocietyExample')
A = getContactFrequency(myhic,19,5000)
head(print(A))
```

`getContactProbability` *Contact probability*

Description

It estimates contact probability based on the distance of a pair of a loci.

Usage

```
getContactProbability(
  tab,
  farthest = 2000000,
  resol = 10000,
  prob,
  n_cores = NULL
)
```

Arguments

`tab`	Output from getContactFrequency function.
`farthest`	Maximum 1-D distance to search. Default=2Mb
`resol`	Hi-C resolution for test. Default = 10000
`prob`	Significance cutoff for negative binomial distribution. Default =0.975
`n_cores`	The number of cores used for parallel computing. If set as NULL, n_cores is automatically set to the number of cores in the computer if it is not exceed 30. If it is more than 30, it is set as 30. Default = NULL

Details

Get Contact probablity

Value

A `list` containing three objects: `AREA`, `original`, and `len1`, representing the statistical significance of each chromatin interaction pair.

Author(s)

Sora Yoon, PhD

Examples

```
# This example might take a long time to run, so we wrap it in donttest{}

myhic = system.file('extdata','example.hic',package = 'HiCocietyExample')
mydf=getContactFrequency(myhic, 19, 5000);
myprob=getContactProbability(mydf,farthest=2000000, resol=5000,prob=0.975,
n_cores=2);
```

`getElbowPoint` *Estimation of elbow point from J-shaped-curve*

Description

It provides a point of the highest curvature from a J-shaped-plot

Usage

```
getElbowPoint(numbers)
```

Arguments

`numbers`	Numeric vector

Details

Estimation of elbow point from J-shaped curve

Value

A `list` containing two elements: `index`, the index of a point in a sorted vector of numbers in descending order, representing the point where the tangent is closest to one, and `ConnectivityCutoff`, the corresponding value at that point.

Author(s)

Sora Yoon, PhD

Examples

```
modulefile = system.file('extdata','mouse_naiveCD4T_Vahedi_short.rds',
package = 'HiCocietyExample')
mycom = readRDS(modulefile)
connec = mycom$ModuleSummary$connectivity
getElbowPoint(connec)
```

`get_all_chr_names` *Retrieve chromosome names from .hic file*

Description

It retrieves all chromosome names having longer than 2.5Mbp.

Usage

```
get_all_chr_names(fname)
```

Arguments

`fname`	Path to .hic file

Details

To extract all chromosome names from .hic file

Value

A character vector containing the names of chromosomes whose genomic lengths exceed 2.5 Mbp.

Author(s)

Sora Yoon, PhD

Examples

```
myhic=system.file('extdata', 'example.hic', package ='HiCocietyExample')
get_all_chr_names(myhic)
```

`get_txdb` *All available Txdb*

Description

It finds all available Txdb packages used in add_Genes function.

Usage

```
get_txdb()
```

Details

Check all available Txdb package

Value

A character vector containing the names of all available TxDb packages.

Author(s)

Sora Yoon, PhD

Examples

```
get_txdb()
```

`hic2community` *Create module objects from the Hi-C data*

Description

It generates a list of graph of significant interactions, module table and module elements.

Usage

```
hic2community(
  fname,
  chr,
  resol,
  nbprob,
  farthest,
  par.noise = 1,
  network.cluster.method = "louvain",
  n_cores = NULL
)
```

Arguments

`fname`	Path to .hic file
`chr`	chromosome numbers to run.
`resol`	Resolution of Hi-C data
`nbprob`	Negative binomial probability. Higher value gives smaller number of stronger interaction.
`farthest`	The maximum searching distance between two nodes
`par.noise`	Parameter for noise removal. Default is 1, higher value gives more filtered interactions.
`network.cluster.method`	Can select between 'louvain' as default and 'label_prop' which means the label propagation method.
`n_cores`	The number of cores used for parallel computing. If set as NULL, n_cores is automatically set to the number of cores in the computer if it is not exceed 30. If it is more than 30, it is set as 30. Default = NULL

Details

It generates a list of graph of significant interactions, module table and module elements.

Value

A `list` containing three elements: `Graphs` (an `igraph` object representing significant chromatin interactions for each chromosome), `ModuleSummary` (a `data.frame` containing information about chromatin interaction modules), and `ModuleElements` (a `list` of nodes forming significant chromatin interactions within each module).

Author(s)

Sora Yoon, PhD

Examples

```
# This example might take a long time to run, so we wrap it in donttest{}

myhic=system.file('extdata', 'example.hic', package ='HiCocietyExample')
mycom = hic2community(myhic, "19", 5000, 0.975, 2000000,
par.noise=1, 'louvain', n_cores=2)
```

`hic2network` *HiC to network data format*

Description

It converts Hi-C dataframe to network object.

Usage

```
hic2network(ftab)
```

Arguments

`ftab`	three-column data composed of locus1, locus2 and value

Details

Convert HiC to network data format

Value

An `igraph` object representing statistically significant chromatin interactions.

Author(s)

Sora Yoon, PhD

Examples

```
# This example might take a long time to run, so we wrap it in donttest{}

myhic=system.file('extdata', 'example.hic', package ='HiCocietyExample')
ftab=getContactFrequency(myhic,19,5000);
net = hic2network(ftab[1:100,]);
plot(net)
```

---

visualizeModule *Visualization of module*

Description

It draws a triangle heatmap and arcplot of a module

Usage

```
visualizeModule(
  hicpath,
  HC.object,
  moduleNum,
  resolution,
  hic.norm,
  heatmap.color.range = NULL,
  heatmap.color = colorRampPalette(c("white", "red")),
  arc.depth = 10,
  arc.color = "gray80",
  nbnom.param = 0.99,
  txdb = "TxDb.Mmusculus.UCSC.mm10.knownGene",
  gene.strand.arrow.lwd = 3,
  gene.strand.lwd = 6,
  col.forward.gene = "purple",
  col.reverse.gene = "pink",
  highlight.centrality = FALSE,
  highlight.cent.col = FALSE,
  highlight.node = NULL,
  highlight.node.col = NULL,
  show.sig.int = TRUE,
  netinfo
)
```

Arguments

`hicpath`	Path to the .hic file
`HC.object`	The object name from hic2community result
`moduleNum`	The row index of module to draw
`resolution`	Resolution of HiC data
`hic.norm`	Normalization method. If not, set 'NONE'
`heatmap.color.range`	Min and max value of contact frequency, e.g., c(0,10)
`heatmap.color`	Color for heatmap. For example, colorRampPalette(c("white","red"))
`arc.depth`	Height of arc plot
`arc.color`	Arc color

`nbnom.param`	Negative binomial probability cutoff. Higher cutoff gives less number of arcs.
`txdb`	Character. One of Txdb list obtained from get_txdb().
`gene.strand.arrow.lwd`	Numeric. Line width of arrowhead indicating the strands of genes. Same as arr.lwd option in Arrows function in shape package.
`gene.strand.lwd`	Numeric. Line width of arrow body indicating the strands of genes. Same as lwd option in Arros function in shape package.
`col.forward.gene`	Character. Color of arrows within gene track for forward genes.
`col.reverse.gene`	Character. Color of arrows within gene track for reverse genes
`highlight.centrality`	Boolean input to set if highlight eigenvector centrality node.
`highlight.cent.col`	The color of arcs stemming from the centrality node.
`highlight.node`	The coordiante of a node of which the user will highlight the arcs stemming from this node. Default=NULL
`highlight.node.col`	The color of arcs stemming from the node which the user highlight.
`show.sig.int`	Boolean. If TRUE, it marks significant contact on the triangle heatmap.
`netinfo`	Boolean. If TRUE, it shows network information of the module as text in the plot.

Details

Visualization of module

Value

No return value; the function generates a plot.

Author(s)

Sora Yoon, PhD

Examples

```
# A slow example that takes too long to run, wrapped in donttest{}

myhic = system.file('extdata','example.hic',package = 'HiCocietyExample')
HC.object = hic2community(myhic, "19", 5000, 0.975, 2000000, par.noise=1,
'louvain', n_cores=2)
mNum = 1
visualizeModule(hicpath = myhic, HC.object = HC.object, moduleNum = mNum,
resolution = 5000,
hic.norm = 'NONE', heatmap.color.range=c(0,10),
heatmap.color = colorRampPalette(c('white','red')),
```

```
arc.depth=10, arc.color = "gray80", nbnom.param=0.99,
txdb = 'TxDb.Mmusculus.UCSC.mm10.knownGene',
gene.strand.arrow.lwd = 3, gene.strand.lwd = 3,
col.forward.gene = 'purple', col.reverse.gene = 'pink',
highlight.centrality=FALSE, highlight.cent.col=FALSE,
highlight.node=NULL, highlight.node.col=NULL,
show.sig.int=FALSE, netinfo=FALSE)
```

Index