Package 'InTAD'

February 19, 2026

Type Package

Title Search for correlation between epigenetic signals and gene expression in TADs

Version 1.30.0

Author Konstantin Okonechnikov, Serap Erkek, Lukas Chavez

Maintainer Konstantin Okonechnikov <k.okonechnikov@gmail.com>

Description The package is focused on the detection of correlation between expressed genes and selected epigenomic signals (i.e. enhancers obtained from ChIP-seq data) either within topologically associated domains (TADs) or between chromatin contact loop anchors. Various parameters can be controlled to investigate the influence of external factors and visualization plots are available for each analysis step.

License GPL (>=2)

LazyData TRUE

Depends R (>= 3.5), methods, S4Vectors, IRanges, GenomicRanges, MultiAssayExperiment, SummarizedExperiment,stats

Imports BiocGenerics,Biobase,rtracklayer,parallel,graphics,mclust,qvalue, ggplot2,utils,ggpubr

biocViews Epigenetics, Sequencing, ChIPSeq, RNASeq, HiC, GeneExpression,ImmunoOncology

VignetteBuilder knitr

Suggests testthat, BiocStyle, knitr, rmarkdown

RoxygenNote 6.0.1

git_url https://git.bioconductor.org/packages/InTAD

git_branch RELEASE_3_22

git_last_commit a8e25c4

git_last_commit_date 2025-10-29

Repository Bioconductor 3.22

Date/Publication 2026-02-18

Contents

combineInTAD 2
combineWithLoops 3

enhSel 4
enhSelGR 4
exprs,InTADSig-method 5
filterGeneExpr 5
findCorFromLoops 6
findCorrelation 7
fnSE 7
geneCoords 8
get.enr.bg.normfit 9
InTADSig 9
loadSigInTAD 10
loopsDfSel 11
mbAnnData 11
newSigInTAD 12
plotCorAcrossRef 13
plotCorrelation 13
rpkmCountsSel 14
sigCoords 15
signals 15
tadGR 16
txsSel 16

Index **17**

`combineInTAD` *Preparation for correlation analysis*

Description

This function combines signals and genes in inside of Topologically Associated Domains (TADs)

Usage

```
combineInTAD(object, tadGR, selMaxTadOvlp = TRUE, closestGene = TRUE)
```

Arguments

`object`	InTADSig object
`tadGR`	TAD genomic regions
`selMaxTadOvlp`	If a signal overlaps 2 or more TADs by default only single TAD with max overlap is selected.All overlaps can be included by deactivating this option.
`closestGene`	By default closest to TAD genes are selected based on TSS location. Deactivate this option to use genes only lying within TAD.

Details

Each signal is checked if it is lying inside of TAD. Signals out of TADs are ignored. The genomic regions reprenting gene coordiantes are converted to TSS. By default, the closest genes are assigned belonging to TAD. If this option deactivated, only those lying with TAD are collected. Result is a list of signals connected to tables with gene details.

Value

Updated InTADSig object containing genes connected to eash signal

Examples

```
# create sigInTAD object
inTadSig <- newSigInTAD(enhSel, enhSelGR, rpkmCountsSel, txsSel)
# combine signals and genes in TAD
inTadSig <- combineInTAD(inTadSig, tadGR)
```

`combineWithLoops` *Preparation for correlation analysis via loops*

Description

This function combines signals and genes based on the usage of loops obtained from HiC data analysis

Usage

```
combineWithLoops(object, loopsInitDf, fragmentLength = 0, tssWidth = 2000,
  extSize = 0)
```

Arguments

`object`	InTADSig object
`loopsInitDf`	Data frame with loops. By default 6-column format *(chr1,start1,end1,chr2,start2,pos2)* is expected.
`fragmentLength`	In case the input format is 4-column *(chr1,middlePos1, chr2, middlePos2)* fragment length should be provided to extend the corresponding loci for loop start and end positions.
`tssWidth`	The transcription start site width is used to control overlaps with loop anchor. Default is 2000 base pairs.
`extSize`	The loop endings can be extended upstream and downstream with provided corresponding increase size in base pairs.

Details

The expected input is the loops data.frame applied to find connections of signals to genes. This data.frame could be in two formats: either *(chr1,start1,end1,chr2,start2,end2)* or *(chr1,middlePos1,chr2,middlePos2)* with fragment size.

Value

Updated InTADSig object containing genes connected to signals via loops

enhSel — *Enhancer signals subset detected from medulloblatoma samples*

Description

This data.frame contains 65 selected in chr15 normalized enhancers signals subset from 25 medulloblastoma samples.

Usage

```
enhSel
```

Format

a data.frame instance

Value

NULL, but makes available the dataframe

enhSelGR — *Genomic coordiantes of enhancer signals subet*

Description

This GRanges object contains the coordinates of 65 medulloblastoma enhancer signals in chr15 target region

Usage

```
enhSelGR
```

Format

a GRanges object

Value

NULL, but makes available the dataset

exprs,InTADSig-method *Gene expression counts table*

Description

This funcion returns gene expression counts table

Usage

```
## S4 method for signature 'InTADSig'
exprs(object)
```

Arguments

object	InTADSig object with signals and genes

Value

Gene expression table

Examples

```
inTadSig <- newSigInTAD(enhSel, enhSelGR, rpkmCountsSel, txsSel)
head(exprs(inTadSig))
```

filterGeneExpr *Function to filter gene expression*

Description

This function performs filtering of gene expression counts based on various parameters

Usage

```
filterGeneExpr(obj, cutVal = 0, geneType = NA, checkExprDistr = FALSE,
  plotExprDistr = FALSE)
```

Arguments

obj	InTADSig object
cutVal	Exclude genes that have max expression less or equal to this value in all samples. Default: 0
geneType	Type of gene to select for filtering i.e. "protein_coding". Default:NA
checkExprDistr	Adjust cutVal based on gene expression distribution
plotExprDistr	Perform visualziation of the distribution

Details

The function allows to stabilize the functional activity of the genes. By default all not expressed genes are filtered. It is also possible to set type of gene to take into account i.e. "protein_coding" only. This option requires additional metadata column "transcript_type". Also, special filtering option based on mclust library allows to analyze distribution of counts and adjust the cut value to exclude low expressed genes.

Value

InTADSig object with filtered counts table

Examples

```
## perform analysis on test data
inTadSig <- newSigInTAD(enhSel, enhSelGR, rpkmCountsSel, txsSel)
## default filtering
inTadSig <- filterGeneExpr(inTadSig)
## filter based on gene type
inTadSig <- filterGeneExpr(inTadSig, geneType = "protein_coding")
```

---

findCorFromLoops *Function to perfrom correlation analysis via loops.*

Description

This function combines genes and signals using obtained loop connections.

Usage

```
findCorFromLoops(object, method = "pearson", adj.pval = FALSE)
```

Arguments

object	InTADSig object with signals and genes combined via loops
method	Correlation method: "pearson" (default), "kendall", "spearman"
adj.pval	Perform p-value adjsutment and include q-values in result

Value

A table with correlation values for signal-gene pairs including correlation p-value and euclidian distance.

findCorrelation *Function to perfrom correlation analysis in TADs*

Description

This function combines genes and signals in inside of TADs

Usage

```
findCorrelation(object, method = "pearson", adj.pval = FALSE,
  plot.proportions = FALSE)
```

Arguments

object	InTADSig object with signals and genes combined in TADS
method	Correlation method: "pearson" (default), "kendall", "spearman"
adj.pval	Perform p-value adjsutment and include q-values in result
plot.proportions	Plot proportions of signals and genes in correlation

Value

A table with correlation values for signal-gene pairs including correlation p-value, euclidian distance and rank.

Examples

```
## perform analysis on test data
inTadSig <- newSigInTAD(enhSel, enhSelGR, rpkmCountsSel, txsSel)
inTadSig <- filterGeneExpr(inTadSig, geneType = "protein_coding")
inTadSig <- combineInTAD(inTadSig, tadGR)
corData <- findCorrelation(inTadSig, method="pearson")
```

fnSE *Preparation for correlation analysis for a signal*

Description

This function collects all genes for signal genomic region inside of Topologically Associated Domains (TADs)

Usage

```
fnSE(id, sigList, tadGR, tss, pickMaxOvlp, nearestTad)
```

Arguments

id	Id of signal from the list
sigList	List of signal GRs and their names
tadGR	TAD genomic regions
tss	Gene transcription start sites
pickMaxOvlp	Use TAD with max overlap
nearestTad	The table listing TADs nearest to each TSS #'

Details

The signal is checked if it is lying inside of TAD. Then all genes in this TAD are collected.

Value

Data.frame containing genes connected to signal

---

geneCoords *Gene coords GRanges*

Description

This funcion returns the gene GRanges

Usage

```
geneCoords(object)

## S4 method for signature 'InTADSig'
geneCoords(object)
```

Arguments

object	InTADSig object with signals and genes

Value

Gene GRanges

Examples

```
inTadSig <- newSigInTAD(enhSel, enhSelGR, rpkmCountsSel, txsSel)
head(geneCoords(inTadSig))
```

get.enr.bg.normfit *Function to estimate gene expression*

Description

This function uses mclust package to analyze gene expression distribution

Usage

```
get.enr.bg.normfit(x)
```

Arguments

x	Full gene expression vector

Details

The function adjust filtering cut value based on mclust library to exclude low expressed genes. It is a part of filtering procedure.

Value

Distribution properties: mean and std

InTADSig *The InTADSig Class*

Description

The InTADSig object stores signals and gene expression data for the samples.

Details

It uses MultiAssayExperiment object to store information. Key slots to access are listed below.

Slots

`sigMAE`: `"MultiAssayExperiment"`, MultiAssayExperiment object containg signals and gene counts

`signalConnections`: `"list"`, The list of signals representing gene data frames in the same TAD

`loopsDf`: `"data.frame"`, The data.frame containing details of provided input loops

`loopConnections`: `"list"`, The list of connections between signals and genes via loops

`ncore`: `"numeric"`, Number of cores to use for parallel computing #'

loadSigInTAD *Load InTADSig object from text files*

Description

The fuction loads the data tables to create an object that contains the signals and gene expression data.frames along with their genomic coordinates for further processing.

Usage

```
loadSigInTAD(signalsFile, countsFile, gtfFile, annFile = "",
  performLog = TRUE, logExprsOffset = 1, ncores = 1)
```

Arguments

signalsFile	Tab-seprated data table containg signals and their coordinates as row.names
countsFile	Tab-seprated counts table
gtfFile	GTF file containing all gene coordinates
annFile	Tab-delimited phenotype annotation of samples
performLog	Perform log2 convertion of expression values. Default: TRUE.
logExprsOffset	Offset x for log2 gene exrpression i.e. log2(value + x). Default: 1
ncores	Number of cores to use for parallel computing

Details

The function loads data from input files and creates object that stores matrices of signals and gene expression values along with coordiantes. The samples order and names of columns should match in both tables. It is expected that gene ids are applied in the validation of counts table.

Value

Novel InTADSig object

Examples

```
# create sigInTAD object
inTadSig <- newSigInTAD(enhSel, enhSelGR, rpkmCountsSel, txsSel)
```

loopsDfSel *Data frame containing coordinates of loops*

Description

The table contains genomic coordinates of chromatin loops in 6-column format derived from IMR90 cell line (focus : chr15)

Usage

```
loopsDfSel
```

Format

a data.frame object

Value

NULL, but makes available the dataset

mbAnnData *Data frame containing information about samples*

Description

The table includes additional informaiton about MB tumour samples (subgroup, gender, age, histology and M.Stage)

Usage

```
mbAnnData
```

Format

a data.frame object

Value

NULL, but makes available the dataset

`newSigInTAD` *Create InTADSig object*

Description

The fuction generates an object that contains the signals and gene expression data.frames along with their genomic coordinates for further processing.

Usage

```
newSigInTAD(signalData = NULL, signalRegions = NULL, countsData = NULL,
  geneRegions = NULL, sampleInfo = NULL, performLog = TRUE,
  logExprsOffset = 1, ncores = 1)
```

Arguments

`signalData`	data frame containing signals
`signalRegions`	genomic regions of the signals
`countsData`	data matrix containing count expression values
`geneRegions`	gene coordiantes
`sampleInfo`	data frame containing additional sample info
`performLog`	Perform log2 convertion of expression values. Default: TRUE.
`logExprsOffset`	Offset x for log2 gene exrpression i.e. log2(value + x). Default: 1
`ncores`	Number of cores to use for parallel computing

Details

InTADSig object stores matrices of signals and gene expression values along with coordinates. The order of samples and names of columns should match in both datasets. For gene coordinates GRanges "gene_id" and "gene_name" are required in metadata. These are typical markers of genes in GTF anntotation format.

Value

Novel InTADSig object

Examples

```
## create sigInTAD object
inTadSig <- newSigInTAD(enhSel, enhSelGR, rpkmCountsSel, txsSel)
```

plotCorAcrossRef *Function to plot correlation across genome*

Description

This function creates a plot of correlation strength in target genomic region from the result table. The X-coordinates represent signals, Y-coords represent genes, while each dot represents -log10(P-value) from correlation test. Additionallly all TAD boundaries can be visualized.

Usage

```
plotCorAcrossRef(obj, corRes, targetRegion, showCorVals = FALSE,
  symmetric = FALSE, tads = NULL)
```

Arguments

obj	InTADSig object with signals and genes combined in TADS
corRes	Correlation result table created by function findCorrelation()
targetRegion	Target genomic region visualise.
showCorVals	Use this option to visualize postive correlation values instead of correlation strength
symmetric	Activate mirrow symmetry for gene-signal connections
tads	TAD regions to visualize. By default only TADs persent in correlation result table are applied (NULL value).

Value

A ggplot object for visualization or customization.

Examples

```
inTadSig <- newSigInTAD(enhSel, enhSelGR, rpkmCountsSel, txsSel)
inTadSig <- combineInTAD(inTadSig, tadGR)
corData <- findCorrelation(inTadSig, method="pearson")
plotCorAcrossRef(inTadSig,corData,GRanges("chr15:25000000-28000000"))
```

plotCorrelation *Function to plot correlation*

Description

This function creates a plot of selected pair signal-gene

Usage

```
plotCorrelation(obj, sId, geneName, xLabel = "Gene expression",
  yLabel = "Signal enrichment", colByPhenotype = "",
  corMethod = "pearson")
```

Arguments

obj	InTADSig object with signals and genes combined in TADS
sId	Signal id based on genomic cooridantes i.e. "chr:start-end"
geneName	Gene name to select. Based on "gene_name" attribute.
xLabel	The label to mark signal X-axis. Default: "Gene expression"
yLabel	The label to mark signal Y-axis. Default: "Signal enrichment"
colByPhenotype	The pheno data column i.e. tumour type that can be use for colour
corMethod	Correlation method. Default: Pearson

Value

A ggplot object for visualization or customization.

Examples

```
inTadSig <- newSigInTAD(enhSel, enhSelGR, rpkmCountsSel, txsSel)
inTadSig <- combineInTAD(inTadSig, tadGR)
plotCorrelation(inTadSig, "chr15:26372163-26398073", "GABRA5")
```

---

rpkmCountsSel *Gene expression subset from medulloblastoma samples*

Description

This data.frame contains RPKM gene expression values from chr15 for subset from 25 medulloblastoma samples.

Usage

```
rpkmCountsSel
```

Format

a data.frame instance

Value

NULL, but makes available the dataframe

sigCoords *Signal coords GRanges*

Description

This funcion returns the signal GRanges

Usage

```
sigCoords(object)

## S4 method for signature 'InTADSig'
sigCoords(object)
```

Arguments

object	InTADSig object with signals and genes

Value

Signal GRanges

Examples

```
inTadSig <- newSigInTAD(enhSel, enhSelGR, rpkmCountsSel, txsSel)
head(sigCoords(inTadSig))
```

signals *Signal values table*

Description

This funcion returns the signal values table

Usage

```
signals(object)

## S4 method for signature 'InTADSig'
signals(object)
```

Arguments

object	InTADSig object with signals and genes

Value

Signals table

Examples

```
inTadSig <- newSigInTAD(enhSel, enhSelGR, rpkmCountsSel, txsSel)
head(signals(inTadSig))
```

tadGR *Genomic coordiantes of topologically associated domains*

Description

This GRanges object contains the coordinates of TADs revealed from IMR90 cell line (extracted from 0-indexed .bed file)

Usage

```
tadGR
```

Format

a GRanges object

Value

NULL, but makes available the dataset

txsSel *Genomic coordiantes of genes subset*

Description

This GRanges object contains the coordinates of genes subset from chr15

Usage

```
txsSel
```

Format

a GRanges object

Value

NULL, but makes avaialbe the dataset

Index

combineInTAD, 2
combineWithLoops, 3

enhSel, 4
enhSelGR, 4
exprs,InTADSig-method, 5

filterGeneExpr, 5
findCorFromLoops, 6
findCorrelation, 7
fnSE, 7

geneCoords, 8
geneCoords,InTADSig-method
(geneCoords), 8
get.enr.bg.normfit, 9

InTADSig, 9
InTADSig-class (InTADSig), 9

loadSigInTAD, 10
loopsDfSel, 11

mbAnnData, 11

newSigInTAD, 12

plotCorAcrossRef, 13
plotCorrelation, 13

rpkmCountsSel, 14

sigCoords, 15
sigCoords,InTADSig-method (sigCoords),
15
signals, 15
signals,InTADSig-method (signals), 15

tadGR, 16
txsSel, 16