Package ‘NoRCE’

July 24, 2025

Type Package

Title NoRCE: Noncoding RNA Sets Cis Annotation and Enrichment

Version 1.21.2

Description While some non-coding RNAs (ncRNAs) are assigned critical regulatory roles, most remain functionally uncharacterized. This presents a challenge whenever an interesting set of ncRNAs needs to be analyzed in a functional context. Transcripts located close-by on the genome are often regulated together. This genomic proximity on the sequence can hint to a functional association. We present a tool, NoRCE, that performs cis enrichment analysis for a given set of ncRNAs. Enrichment is carried out using the functional annotations of the coding genes located proximal to the input ncRNAs. Other biologically relevant information such as topologically associating domain (TAD) boundaries, co-expression patterns, and miRNA target prediction information can be incorporated to conduct a richer enrichment analysis. To this end, NoRCE includes several relevant datasets as part of its data repository, including cell-line specific TAD boundaries, functional gene sets, and expression data for coding & ncRNAs specific to cancer. Additionally, the users can utilize custom data files in their investigation. Enrichment results can be retrieved in a tabular format or visualized in several different ways. NoRCE is currently available for the following species: human, mouse, rat, zebrafish, fruit fly, worm, and yeast.

License MIT + file LICENSE

Depends R (>= 4.4.0)

Imports KEGGREST,png,dplyr,graphics,RSQLite,DBI,tidyr,grDevices,stringr,Seqinfo,
S4Vectors,SummarizedExperiment,reactome.db,rWikiPathways,RCurl,
dbplyr,utils,ggplot2,igraph,stats,reshape2,readr, GO.db,
biomaRt,rtracklayer,IRanges,GenomicRanges,GenomicFeatures,AnnotationDbi,methods

Encoding UTF-8

RoxygenNote 7.3.2

Suggests knitr,
TxDb.Hsapiens.UCSC.hg38.knownGene,TxDb.Drerio.UCSC.danRer10.refGene,
TxDb.Mmusculus.UCSC.mm10.knownGene,TxDb.Dmelanogaster.UCSC.dm6.ensGene,
testthat,TxDb.Celegans.UCSC.ce11.refGene,rmarkdown,
TxDb.Rnorvegicus.UCSC.rn6.refGene,TxDb.Hsapiens.UCSC.hg19.knownGene,
org.Mm.eg.db,
org.Rn.eg.db,org.Hs.eg.db,org.Dr.eg.db,BiocGenerics,
org.Sc.sgd.db, org.Ce.eg.db,org.Dm.eg.db, markdown

VignetteBuilder knitr

biocViews BiologicalQuestion, DifferentialExpression,
GenomeAnnotation, GeneSetEnrichment, GeneTarget,
GenomeAssembly, GO

LazyData true

BugReports https://github.com/guldenolgun/NoRCE/issues

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

git_branch devel

git_last_commit 13714c3

git_last_commit_date 2025-07-22

Repository Bioconductor 3.22

Date/Publication 2025-07-23

Author Gulden Olgun [aut, cre]

Maintainer Gulden Olgun <gulden@cs.bilkent.edu.tr>

Contents

`annGO` *Annotate the set of genes with the GO terms for a given species and assembly*

Description

Annotate the set of genes with the GO terms for a given species and assembly

Usage

```
annGO(
  genes,
  GOtype = c("BP", "CC", "MF"),
  org_assembly = c("hg19", "hg38", "mm10", "dre10", "rn6", "dm6", "ce11", "sc3")
)
```

Arguments

`genes`	List of mRNA genes. Supported format for genes is Hugo.
`GOtype`	Hierarchical category of the GO ontology. Possible values are 'BP', 'CC', 'MF'.
`org_assembly`	Genome assembly of interest. Possible assemblies are 'mm10' for mouse, 'dre10' for zebrafish, 'rn6' for rat, 'dm6' for fruit fly, 'ce11' for worm, 'hg19' and 'hg38' for human

Value

data frame of the GO term annotation of the genes

`assembly` *Get the required information for the given assembly*

Description

Get the required information for the given assembly

Usage

```
assembly(
  org_assembly = c("hg19", "hg38", "mm10", "dre10", "rn6", "dm6", "ce11", "sc3")
)
```

Arguments

`org_assembly`	Genome assembly of interest for the analysis. Possible assemblies are "mm10" for mouse, "dre10" for zebrafish, "rn6" for rat, "dm6" for fruit fly, "ce11" for worm, "sc3" for yeast, "hg19" and "hg38" for human

Value

setting required information

Examples

```
 ## Not run:
assembly('hg19')

## End(Not run)
```

`brain_disorder_ncRNA` *Differentially expressed non-coding gene*

Description

Differentially expressed non-coding gene

Usage

```
brain_disorder_ncRNA
```

Format

Not Available

Source

http://resource.psychencode.org/

Examples

```
data(brain_disorder_ncRNA)
```

`brain_mirna` *Differentially expressed human brain data*

Description

Differentially expressed human brain data

Usage

```
brain_mirna
```

Format

Not Available

Source

http://resource.psychencode.org/

Examples

```
data(brain_mirna)
```

`breastmRNA` *Protein coding genes that are differentially expressed in TCGA breast cancer RNAseq data.*

Description

Protein coding genes that are differentially expressed in TCGA breast cancer RNAseq data.

Usage

```
breastmRNA
```

Format

Not Available

Source

https://portal.gdc.cancer.gov/

Examples

```
data(breastmRNA)
```

`calculateCorr` *Calculates the correlation coefficient values between two custom expression data.*

Description

Calculates the correlation coefficient values between two custom expression data.

Usage

```
calculateCorr(
  exp1,
  exp2,
  label1 = "",
  label2 = "",
  corrMethod = "pearson",
  varCutoff = 0.0025,
  corCutoff = 0.3,
  pcut = 0.05,
  alternate = "greater",
  conf = 0.95
)
```

Arguments

`exp1`	Custom expression data matrix or SummarizedExperiment data. Columns must be genes and rows must be patients.
`exp2`	Custom expression data matrix or SummarizedExperiment data. Columns must be genes and rows must be patients.
`label1`	Gene names of the custom exp1 expression data. If it is not provided, column name of the exp1 data will be taken.
`label2`	Gene names of the custom exp2 expression data. If it is not provided, column name of the exp2 data will be taken.
`corrMethod`	Correlation coeffient method that will be used for evaluation. Possible values are "pearson", "kendall", "spearman"
`varCutoff`	Variance cut off that genes have less variance than this value will be trimmed
`corCutoff`	Correlation cut off values for the given correlation method
`pcut`	P-value cut off for the correlation values
`alternate`	Holds the alternative hypothesis and "two.sided", "greater" or "less" are the possible values.
`conf`	Confidence level for the returned confidence interval. It is only used for the Pearson correlation coefficient if there are at least 4 complete pairs of observations.

Value

Pairwise relations between gene-gene with corresponding correlation value and pvalue

Examples

```
## Not run:
#Assume that mirnanorce and mrnanorce are custom patient by gene data
a<-calculateCorr(exp1 = mirna, exp2 = mrna )

## End(Not run)
```

---

convertGeneID *Convert gene ids according to the gene type*

Description

Convert gene ids according to the gene type

Usage

```
convertGeneID(
  genetype = c("Entrez", "mirna", "Ensembl_gene", "Ensembl_trans", "NCBI", "All"),
   genelist,
   org_assembly = c("hg19", "hg38", "mm10", "dre10", "rn6", "dm6", "ce11", "sc3")
)
```

Arguments

`genetype`	Type of the input gene list. Possible values are "Entrez", "mirna", "Ensembl_gene", "Ensembl_trans", "NCBI". For HUGO gene symbol "NCBI" value, for Entrez gene id "Entrez", for mirbase id "mirna" is used.
`genelist`	Input gene list
`org_assembly`	Genome assembly of interest for the analysis. Possible assemblies are "mm10" for mouse, "dre10" for zebrafish, "rn6" for rat, "dm6" for fruit fly, "ce11" for worm, "sc3" for yeast, "hg19" and "hg38" for human

Value

GRange object of the given input

Examples

```
## Not run:
convGene <-convertGeneID(genetype = "mirna",
                          genelist = brain_mirna[1:30,],
                          org_assembly = 'hg19')

## End(Not run)
```

`convertGMT` *Convert gmt formatted pathway file to the Pathway ID, Entrez, symbol formatted data frame*

Description

Convert gmt formatted pathway file to the Pathway ID, Entrez, symbol formatted data frame

Usage

```
convertGMT(gmtName, org_assembly, isSymbol = FALSE)
```

Arguments

`gmtName`	Custom pathway gmt file
`org_assembly`	Genome assembly of interest for the analysis. Possible assemblies are "mm10" for mouse, "dre10" for zebrafish, "rn6" for rat, "dm6" for fruit fly, "ce11" for worm, "sc3" for yeast, "hg19" and "hg38" for human
`isSymbol`	Boolean variable that hold the gene format of the gmt file. If it is set as TRUE, gene format of the gmt file should be symbol. Otherwise, gene format should be ENTREZ ID. By default, it is FALSE.

Value

return data frame

`corrbased` *Pearson correlation coefficient value of the miRNA genes between miRNA:mRNA for a given correlation cut-off and cancer.*

Description

Pearson correlation coefficient value of the miRNA genes between miRNA:mRNA for a given correlation cut-off and cancer.

Usage

```
corrbased(mirnagene, cancer, minAbsCor, databaseFile)
```

Arguments

`mirnagene`	Data frame of the miRNA genes in mature format
`cancer`	Name of the TCGA project code such as 'BRCA' that is analyzed for miRNA-mRNA correlation. Possible cancer types ACC, BLCA, BRCA, CESC, CHOL, COAD, COADREAD, DLBC, ESCA, GBMLGG, HNSC, KICH, KIPAN, KIRC, KIRP, LGG, LIHC, LUAD, LUSC, OV, PAAD, PCPG, PRAD, READ, SARC, SKCM, STAD, STES, TGCT, THCA, THYM, UCEC, UCS, UVM
`minAbsCor`	Cut-off value for the Pearson correlation coefficient of the miRNA-mRNA
`databaseFile`	Path of the miRcancer.db file

Value

Data frame of the miRNA-mRNA correlation result

`corrbasedMrna` *Pearson correlation coefficient value of the mRNA genes between miRNA:mRNA for a given correlation cut-off and cancer.*

Description

Pearson correlation coefficient value of the mRNA genes between miRNA:mRNA for a given correlation cut-off and cancer.

Usage

```
corrbasedMrna(mRNAgene, cancer, minAbsCor, databaseFile)
```

Arguments

`mRNAgene`	Data frame of the mRNA genes
`cancer`	Name of the TCGA project code such as 'BRCA' that is analyzed for miRNA-mRNA correlation. Possible cancer types ACC, BLCA, BRCA, CESC, CHOL, COAD, COADREAD, DLBC, ESCA, GBMLGG, HNSC, KICH, KIPAN, KIRC, KIRP, LGG, LIHC, LUAD, LUSC, OV, PAAD, PCPG, PRAD, READ, SARC, SKCM, STAD, STES, TGCT, THCA, THYM, UCEC, UCS, UVM
`minAbsCor`	Cut-off value for the Pearson correlation coefficient of the miRNA-mRNA
`databaseFile`	Path of miRcancer.db file

Value

Data frame of the miRNA-mRNA correlation result

`createNetwork` *Create interaction network for top n enriched GO term:coding RNA or GO-term:noncoding RNA interaction. Nodes are GO term and RNA, edges are interactions between them. Each GO-term is annotated and enriched with the mRNAs provided from the input list.*

Description

Create interaction network for top n enriched GO term:coding RNA or GO-term:noncoding RNA interaction. Nodes are GO term and RNA, edges are interactions between them. Each GO-term is annotated and enriched with the mRNAs provided from the input list.

Usage

```
createNetwork(
  mrnaObject,
  type = "pvalue",
  n,
  isNonCode = FALSE,
  takeID = FALSE
)
```

Arguments

`mrnaObject`	Output of enrichment results
`type`	Sort in terms of p-values or FDR. Possible values "pvalue", "padjust"
`n`	Number of top enrichments
`isNonCode`	Boolean value that checks whether node of the network is GO-term for coding or GO-term for noncoding genes. By default, it is FALSE so node of the network is GO-term\& coding gene. Otherwise, nodes are GO-term for noncoding genes.
`takeID`	Boolean value that checks the name decision of the GO/pathway node, GO-term/pathway-term or GO ID-pathway ID. If it is true, name of the GO/pathway node will be GO ID/pathway ID will be used, otherwise, name of the GO/pathway node is GO-term. By default, it is FALSE. It is suggested to used when the GO-term is two long or the GO-term is missing for the custom enrichment database.

Value

Network

`drawDotPlot` *Draw dot plot of the enrichment object*

Description

Draw dot plot of the enrichment object

Usage

```
drawDotPlot(mrnaObject, type = "pAdjust", n)
```

Arguments

`mrnaObject`	Object of the enrichment result
`type`	Draw the dot plot according to the p-value or adjusted p-value ("pvalue", "pAdjust")
`n`	Number of GO terms or pathways, that ordered by type and has least number of top p-value

Value

Dot plot of the top n enrichment results

`extractBiotype` *Get the biotype of the non-coding genes. It is suitable for the GENCODE gtf files*

Description

Get the biotype of the non-coding genes. It is suitable for the GENCODE gtf files

Usage

```
extractBiotype(gtfFile)
```

Arguments

`gtfFile`	Path of the input gtf file which contains biotype information. The gtf file must be provided from the Ensembl or Gencode site. For space efficiency, gft files should be in a zip format.

Value

Tabular form of the gtf file with the required features such as gene id and biotypes

Examples

```
## Not run:
fileImport<-system.file("extdata", "temp.gtf", package = "NoRCE")
gtf <- extractBiotype(gtfFile = fileImport)

## End(Not run)
```

`filterBiotype` *Extract the genes that have user provided biotypes. This method is useful when input gene list is mixed or when research of the interest is only focused on specific group of genes.*

Description

Extract the genes that have user provided biotypes. This method is useful when input gene list is mixed or when research of the interest is only focused on specific group of genes.

Usage

```
filterBiotype(gtfFile, biotypes)
```

Arguments

`gtfFile`	Input gtf file for the genes provided by the extractBiotype function
`biotypes`	Selected biotypes for the genes

Value

Table format of genes with a given biotypes

Examples

```
## Not run:
biotypes <- c('unprocessed_pseudogene','transcribed_unprocessed_pseudogene')
fileImport<-system.file("extdata", "temp.gtf", package = "NoRCE")
extrResult <- filterBiotype(fileImport, biotypes)

## End(Not run)
```

`geneGOEnricher` *Given genes that fall in a given upstream and downstream region of mRNAs of interest, GO term enrichment analysis is carried out*

Description

Given genes that fall in a given upstream and downstream region of mRNAs of interest, GO term enrichment analysis is carried out

Usage

```
geneGOEnricher(
  gene,
 org_assembly = c("hg19", "hg38", "mm10", "dre10", "rn6", "dm6", "ce11", "sc3"),
  genetype = c("Entrez", "mirna", "Ensembl_gene", "Ensembl_trans", "NCBI"),
  backG = "",
  backGType = "pc_gene",
  near = FALSE,
  isTADSearch = FALSE,
  TAD = c(tad_hg19, tad_dmel, tad_hg38, tad_mm10),
  express = FALSE,
  isCustomExp = FALSE,
  cancer,
  exp1,
  exp2,
  label1 = "",
  label2 = "",
  isUnionCorGene = FALSE,
  databaseFile
)
```

Arguments

`gene`	Input genes other than miRNA
`org_assembly`	Genome assembly of interest for the analysis. Possible assemblies are "mm10" for mouse, "dre10" for zebrafish, "rn6" for rat, "dm6" for fruit fly, "ce11" for worm, "sc3" for yeast, "hg19" and "hg38" for human

`genetype`	Type of the input gene list. Possible values are "Entrez", "mirna", "Ensembl_gene", "Ensembl_trans", "NCBI". For HUGO gene symbol "NCBI" value, for Entrez gene id "Entrez" is used.
`backG`	The set of genes that tested against to the input (background gene)
`backGType`	Type of the background gene. If miRNA gene set is used for background gene, backGType should be set to the ’mirna’
`near`	Boolean value presents whether cis-neighbourhood should be considered in the analysis
`isTADSearch`	Boolean value that shows whether TAD analysis is performed. This value has to be TRUE for TAD analysis.
`TAD`	TAD genomic regions for the species. Predefined TAD regions or any new TAD regions can be used for the analysis. TAD regions must be formated as GRanges object. Predefined TAD regions are ’tad_hg19’, ’tad_hg38’, ’tad_mm10’, ’tad_dmel’ for hg19, hg38, mm9 and dm6 assembly, respectively.
`express`	Boolean variable whether co-expression analysis is performed. If this option is set to TRUE, co-expression analysis will be performed.
`isCustomExp`	Boolean variable whether co-expression analysis with custom data will be performed. When this option is set, exp1 and exp2 parameters must be defined.
`cancer`	Defines the name of the TCGA project code such as ’BRCA’ for correlation analysis. Possible cancer types ACC, BLCA, BRCA, CESC, CHOL, COAD, COADREAD, DLBC, ESCA, GBMLGG, HNSC, KICH, KIPAN, KIRC, KIRP, LGG, LIHC, LUAD, LUSC, OV, PAAD, PCPG, PRAD, READ, SARC, SKCM, STAD, STES, TGCT, THCA, THYM, UCEC, UCS, UVM
`exp1`	Custom expression data matrix. Columns must be genes and rows must be patients. If gene names are provided as header, no need to redefine the headers(labels) of the expression data.
`exp2`	Custom expression data matrix. Columns must be genes and rows must be patients. If gene names are provided as header, no need to redefine the headers(labels) of the expression data.
`label1`	Gene names of the custom exp1 expression data. If it is not provided, column name of the exp1 data will be taken.
`label2`	Gene names of the custom exp2 expression data. If it is not provided, column name of the exp2 data will be taken.
`isUnionCorGene`	Boolean value that shows whether union of the output of the co-expression analysis and the other analysis should be considered
`databaseFile`	Path of miRcancer.db file

Value

GO term enrichment object for the given input

Examples

```
## Not run:
ncGO<-geneGOEnricher(gene = brain_disorder_ncRNA, org_assembly='hg19',
near=TRUE, genetype = 'Ensembl_gene')

## End(Not run)
```

---

genePathwayEnricher *Given genes that fall in the given upstream and downstream region of mRNAs of interest, pathway enrichment analysis is carried out*

Description

Given genes that fall in the given upstream and downstream region of mRNAs of interest, pathway enrichment analysis is carried out

Usage

```
genePathwayEnricher(
  gene,
  org_assembly = c("hg19", "hg38", "mm10", "dre10", "rn6", "dm6", "ce11", "sc3"),
  genetype = c("Entrez", "mirna", "Ensembl_gene", "Ensembl_trans", "NCBI"),
  near = TRUE,
  isTADSearch = FALSE,
  TAD = tad_hg19,
  gmtName = "",
  express = FALSE,
  isCustomExp = FALSE,
  cancer,
  exp1,
  exp2,
  label1 = "",
  label2 = "",
  isUnionCorGene = FALSE,
  databaseFile,
  isGeneEnrich = FALSE
)
```

Arguments

`gene`	Input noncoding genes other than miRNA
`org_assembly`	Genome assembly of interest for the analysis. Possible assemblies are "mm10" for mouse, "dre10" for zebrafish, "rn6" for rat, "dm6" for fruit fly, "ce11" for worm, "sc3" for yeast, "hg19" and "hg38" for human
`genetype`	Type of the input gene list. Possible values are "Entrez", "mirna", "Ensembl_gene", "Ensembl_trans", "NCBI". For HUGO gene symbol "NCBI" value, for Entrez gene id "Entrez", for mirbase id "mirna" is used.
`near`	Boolean value presents whether cis-neighbourhood should be considered in the analysis
`isTADSearch`	Boolean value that shows whether TAD analysis is performed. This value has to be TRUE for TAD analysis.
`TAD`	TAD genomic regions for the species. Predefined TAD regions or any new TAD regions can be used for the analysis. TAD regions must be formated as GRanges object. Predefined TAD regions are 'tad_hg19', 'tad_hg38', 'tad_mm10', 'tad_dmel' for hg19, hg38, mm9 and dm6 assembly, respectively.
`gmtName`	Custom pathway gmt file

`express`	Boolean variable whether co-expression analysis is performed. If this option is set to TRUE, co-expression analysis will be performed.
`isCustomExp`	Boolean variable whether co-expression analysis with custom data will be performed. When this option is set, exp1 and exp2 parameters must be defined.
`cancer`	Defines the name of the TCGA project code such as 'BRCA' for correlation analysis. Possible cancer types ACC, BLCA, BRCA, CESC, CHOL,COAD, COADREAD, DLBC, ESCA, GBMLGG, HNSC, KICH, KIPAN, KIRC, KIRP, LIHC, LUAD, LUSC, OV, PAAD, PCPG, PRAD, READ, SARC, SKCM, STAD, STES, TGCT, THCA, THYM, UCEC, UCS, UVM, LGG
`exp1`	Custom expression data matrix. Columns must be genes and rows must be patients. If gene names are provided as header, no need to redefine the headers(labels) of the expression data.
`exp2`	Custom expression data matrix. Columns must be genes and rows must be patients. If gene names are provided as header, no need to redefine the headers(labels) of the expression data.
`label1`	Gene names of the custom exp1 expression data. If it is not provided, column name of the exp1 data will be taken.
`label2`	Gene names of the custom exp2 expression data. If it is not provided, column name of the exp2 data will be taken.
`isUnionCorGene`	Boolean value that shows whether union of the output of the co-expression analysis and the other analysis should be considered
`databaseFile`	Path of miRcancer.db file
`isGeneEnrich`	Boolean value whether gene enrichment should be performed

Value

Pathway enrichment object for the given input

Examples

```
## Not run:
#Pathway enrichment based on the gen sets that falls into the TAD regions
ncRNAPathway<-genePathwayEnricher(gene = brain_disorder_ncRNA ,
                                  org_assembly='hg19',
                                  isTADSearch = TRUE,
                                  TAD = tad_hg19,
                                  genetype = 'Ensembl_gene')
## End(Not run)
```

`geneRegionGOEnricher` *Given gene regions that fall in the given upstream and downstream region of mRNAs of interest, GO term enrichment analysis is carried out*

Description

Given gene regions that fall in the given upstream and downstream region of mRNAs of interest, GO term enrichment analysis is carried out

Usage

```
geneRegionGOEnricher(
  region,
  org_assembly = c("hg19", "hg38", "mm10", "dre10", "rn6", "dm6", "ce11", "sc3"),
  near = TRUE,
  backG = "",
  backGType = "pc_gene",
  isTADSearch = FALSE,
  TAD = c(tad_hg19, tad_dmel, tad_hg38, tad_mm10),
  express = FALSE,
  isCustomExp = FALSE,
  cancer,
  exp1,
  exp2,
  label1 = "",
  label2 = "",
  isUnionCorGene = FALSE,
  databaseFile
)
```

Arguments

`region`	Bed format of the input gene regions other than miRNA
`org_assembly`	Genome assembly of interest for the analysis. Possible assemblies are "mm10" for mouse, "dre10" for zebrafish, "rn6" for rat, "dm6" for fruit fly, "ce11" for worm, "sc3" for yeast, "hg19" and "hg38" for human
`near`	Boolean value presents whether cis-neighbourhood should be considered in the analysis
`backG`	The set of genes that tested against to the input (background gene)
`backGType`	Type of the background gene. If miRNA gene set is used for background gene, backGType should be set to the 'mirna'
`isTADSearch`	Boolean value that shows whether TAD analysis is performed. This value has to be TRUE for TAD analysis.
`TAD`	TAD genomic regions for the species. Predefined TAD regions or any new TAD regions can be used for the analysis. TAD regions must be formated as GRanges object. Predefined TAD regions are 'tad_hg19', 'tad_hg38', 'tad_mm10', 'tad_dmel' for hg19, hg38, mm9 and dm6 assembly, respectively.
`express`	Boolean variable whether co-expression analysis is performed. If this option is set to TRUE, co-expression analysis will be performed.
`isCustomExp`	Boolean variable whether co-expression analysis with custom data will be performed. When this option is set, exp1 and exp2 parameters must be defined.
`cancer`	Defines the name of the TCGA project code such as 'BRCA' for correlation analysis. Possible cancer types ACC, BLCA, BRCA, CESC, CHOL, COAD, COADREAD, DLBC, ESCA, GBMLGG, HNSC, KICH, KIPAN, KIRC, KIRP, LGG, LIHC, LUAD, LUSC, OV, PAAD, PCPG, PRAD, READ, SARC, SKCM, STAD, STES, TGCT, THCA, THYM, UCEC, UCS, UVM
`exp1`	Custom expression data matrix. Columns must be genes and rows must be patients. If gene names are provided as header, no need to redefine the headers(labels) of the expression data.

`exp2`	Custom expression data matrix. Columns must be genes and rows must be patients. If gene names are provided as header, no need to redefine the headers(labels) of the expression data.
`label1`	Gene names of the custom exp1 expression data. If it is not provided, column name of the exp1 data will be taken.
`label2`	Gene names of the custom exp2 expression data. If it is not provided, column name of the exp2 data will be taken.
`isUnionCorGene`	Boolean value that shows whether union of the output of the co-expression analysis and the other analysis should be considered
`databaseFile`	Path of miRcancer.db file

Value

GO term enrichment object for the given input

Examples

```
## Not run:
regions<-system.file("extdata", "ncRegion.txt", package = "NoRCE")
regionNC <- rtracklayer::import(regions, format = "BED")
regionGO<-geneRegionGOEnricher(region = regionNC, org_assembly= 'hg19',
                                near = TRUE)

## End(Not run)
```

---

geneRegionPathwayEnricher

Given gene regions that fall in the given upstream and downstream region of mRNAs of interest, pathway enrichment analysis is carried out

Description

Given gene regions that fall in the given upstream and downstream region of mRNAs of interest, pathway enrichment analysis is carried out

Usage

```
geneRegionPathwayEnricher(
  region,
  org_assembly = c("hg19", "hg38", "mm10", "dre10", "rn6", "dm6", "ce11", "sc3"),
  near = FALSE,
  isTADSearch = FALSE,
  TAD = tad_hg19,
  gmtName = "",
  express = FALSE,
  isCustomExp = FALSE,
  cancer,
  exp1,
  exp2,
```

```
  label1 = "",
  label2 = "",
  isUnionCorGene = FALSE,
  databaseFile,
  isGeneEnrich = FALSE
)
```

Arguments

`region`	Bed format of input gene regions other than miRNA. Input must be Granges object.
`org_assembly`	Genome assembly of interest for the analysis. Possible assemblies are "mm10" for mouse, "dre10" for zebrafish, "rn6" for rat, "dm6" for fruit fly, "ce11" for worm, "sc3" for yeast, "hg19" and "hg38" for human
`near`	Boolean value presents whether cis-neighbourhood should be considered in the analysis
`isTADSearch`	Boolean value that shows whether TAD analysis is performed. This value has to be TRUE for TAD analysis.
`TAD`	TAD genomic regions for the species. Predefined TAD regions or any new TAD regions can be used for the analysis. TAD regions must be formated as GRanges object. Predefined TAD regions are 'tad_hg19', 'tad_hg38', 'tad_mm10', 'tad_dmel' for hg19, hg38, mm9 and dm6 assembly, respectively.
`gmtName`	Custom pathway gmt file
`express`	Boolean variable whether co-expression analysis is performed. If this option is set to TRUE, co-expression analysis will be performed.
`isCustomExp`	Boolean variable whether co-expression analysis with custom data will be performed. When this option is set, exp1 and exp2 parameters must be defined.
`cancer`	Defines the name of the TCGA project code such as 'BRCA' for correlation analysis. Possible cancer types ACC, BLCA, BRCA, CESC, CHOL, COAD, COADREAD, DLBC, ESCA, GBMLGG, HNSC, KICH, KIPAN, KIRC, KIRP, LGG, LIHC, LUAD, LUSC, OV, PAAD, PCPG, PRAD, READ, SARC, SKCM, STAD, STES, TGCT, THCA, THYM, UCEC, UCS, UVM
`exp1`	Custom expression data matrix. Columns must be genes and rows must be patients. If gene names are provided as header, no need to redefine the headers(labels) of the expression data.
`exp2`	Custom expression data matrix. Columns must be genes and rows must be patients. If gene names are provided as header, no need to redefine the headers(labels) of the expression data.
`label1`	Gene names of the custom exp1 expression data. If it is not provided, column name of the exp1 data will be taken.
`label2`	Gene names of the custom exp2 expression data. If it is not provided, column name of the exp2 data will be taken.
`isUnionCorGene`	Boolean value that shows whether union of the output of the co-expression analysis and the other analysis should be considered
`databaseFile`	Path of miRcancer.db file
`isGeneEnrich`	Boolean value whether gene enrichment should be performed

Value

Pathway enrichment object of the given input

Examples

```
## Not run:
regions<-system.file("extdata", "ncRegion.txt", package = "NoRCE")
regionNC <- rtracklayer::import(regions, format = "BED")
ncPath<-geneRegionPathwayEnricher(region = regionNC,
                                  org_assembly = 'hg19',
                                  near = TRUE)
## End(Not run)
```

`getGoDag` *Plot and save the GO term DAG of the top n enrichments in terms of p-values or adjusted p-values with an user provided format*

Description

Plot and save the GO term DAG of the top n enrichments in terms of p-values or adjusted p-values with an user provided format

Usage

```
getGoDag(
  mrnaObject,
  type,
  n,
  filename,
  imageFormat,
  p_range = seq(0, 0.05, by = 0.001)
)
```

Arguments

`mrnaObject`	Output of enrichment results
`type`	Sort in terms of p-values or FDR. possible values "pvalue", "padjust"
`n`	Number of top enrichments
`filename`	Name of the DAG file
`imageFormat`	Image format of the DAG. possible values "png" or "svg"
`p_range`	Break points for the p-values or FDR. By default [0.05, 0.001, 0.0005, 0.0001, 0.00005,0.00001,0] is used

Value

Saves image file in a given format

Examples

```
## Not run:
ncRNAPathway<-mirnaPathwayEnricher(gene = brain_mirna,
                                   org_assembly = 'hg19',near = TRUE)

## End(Not run)
```

---

getKeggDiagram

Display the enriched KEGG diagram of the KEGG pathway. This function is specific to only one KEGG pathway id and identifies the enriched genes in the diagram.

Description

Display the enriched KEGG diagram of the KEGG pathway. This function is specific to only one KEGG pathway id and identifies the enriched genes in the diagram.

Usage

```
getKeggDiagram(
  mrnaObject,
  pathway,
  org_assembly = c("hg19", "hg38", "mm10", "dre10", "rn6", "dm6", "ce11", "sc3")
)
```

Arguments

`mrnaObject`	Output of enrichment results
`pathway`	Kegg pathway term such as 'hsa04010'
`org_assembly`	Genome assembly of interest for the analysis. Possible assemblies are "mm10" for mouse, "dre10" for zebrafish, "rn6" for rat, "dm6" for fruit fly, "ce11" for worm, "sc3" for yeast, "hg19" and "hg38" for human

Value

Shows kegg diagram marked with an enriched genes in a browser

Examples

```
## Not run:
ncRNAPathway<-mirnaPathwayEnricher(gene = brain_mirna,
                                   org_assembly = 'hg19',near = TRUE)

getKeggDiagram(mrnaObject = ncRNAPathway, org_assembly ='hg19',
               pathway = ncRNAPathway@ID[1])

## End(Not run)
```

---

getmiRNACount

Get TCGA miRNAseq expression of miRNA genes for the given cancer

---

Description

Get TCGA miRNAseq expression of miRNA genes for the given cancer

Usage

```
getmiRNACount(mirnagene, cancer, databaseFile)
```

Arguments

`mirnagene`	Data frame of the mature format
`cancer`	Name of the TCGA project code such as 'BRCA'
`databaseFile`	Path of miRcancer.db file

Value

Data frame of the raw read count of the given miRNA genes for different patients

---

`getNearToExon` *Get only those neighbouring genes that fall within exon region*

Description

Get only those neighbouring genes that fall within exon region

Usage

```
getNearToExon(
  bedfile,
  upstream,
  downstream,
  org_assembly = c("hg19", "hg38", "mm10", "dre10", "rn6", "dm6", "ce11", "sc3")
)
```

Arguments

`bedfile`	Input bed formated file
`upstream`	Maximum upstream distance from the TSS position
`downstream`	Maximum downstream distance from the TES position
`org_assembly`	genomee assembly of interest for the analysis. Possible assemblies are "mm10" for mouse, "dre10" for zebrafish, "rn6" for rat, "dm6" for fruit fly, "ce11" for worm, "sc3" for yeast, "hg19" and "hg38" for human

Value

genes

Examples

```
## Not run:
regions <- system.file("extdata", "ncRegion.txt", package = "NoRCE")
regionNC <- rtracklayer::import(regions, format = "BED")

r<-getNearToExon(bedfile = regionNC,
                 upstream = 1000,
                 downstream = 2000,
                 org_assembly = 'hg19')

## End(Not run)
```

`getNearToIntron` *Get only those neighbouring genes that fall within intron region*

Description

Get only those neighbouring genes that fall within intron region

Usage

```
getNearToIntron(
  bedfile,
  upstream,
  downstream,
  org_assembly = c("hg19", "hg38", "mm10", "dre10", "rn6", "dm6", "ce11", "sc3")
)
```

Arguments

`bedfile`	Bed file
`upstream`	upstream distance
`downstream`	downstream distance
`org_assembly`	genomee assembly of interest for the analysis. Possible assemblies are "mm10" for mouse, "dre10" for zebrafish, "rn6" for rat, "dm6" for fruit fly, "ce11" for worm, "sc3" for yeast, "hg19" and "hg38" for human

Value

genes

Examples

```
## Not run:
regions<-system.file("extdata", "ncRegion.txt", package = "NoRCE")
regionNC <- rtracklayer::import(regions, format = "BED")

r<-getNearToExon(bedfile = regionNC,
                 upstream = 1000,
                 downstream = 2000,
                 org_assembly = 'hg19')

## End(Not run)
```

`getReactomeDiagram` *Display the enriched Reactome diagram of the given Reactome pathway id. This function is specific to only one pathway id and identifies the enriched genes in the diagram.*

Description

Display the enriched Reactome diagram of the given Reactome pathway id. This function is specific to only one pathway id and identifies the enriched genes in the diagram.

Usage

```
getReactomeDiagram(mrnaObject, pathway, imageFormat)
```

Arguments

`mrnaObject`	Output of enrichment results
`pathway`	Reactome pathway term
`imageFormat`	Image format of the diagram. Possible image formats are 'png', 'svg'

Value

Shows reactome diagram marked with an enriched genes in a browser

Examples

```
## Not run:
br_enr<-reactomeEnrichment(genes = breastmRNA,org_assembly='hg19')

getReactomeDiagram(mrnaObject = br_enr,pathway = br_enr@ID[1],
                   imageFormat = 'png')

## End(Not run)
```

`getTADOverlap` *For given region of interest, overlapped genes in the TAD regions are found. Results can be filtered according to the available cell lines.*

Description

For given region of interest, overlapped genes in the TAD regions are found. Results can be filtered according to the available cell lines.

Usage

```
getTADOverlap(
  bedfile,
  org_assembly = c("hg19", "hg38", "mm10", "dre10", "rn6", "dm6", "ce11", "sc3"),
  tad = c(tad_hg19, tad_dmel, tad_hg38, tad_mm10),
  near = FALSE,
  upstream = 10000,
  downstream = 10000,
  cellline = "all"
)
```

Arguments

`bedfile`	Region of interest
`org_assembly`	Genome assembly of interest for the analysis. Possible assemblies are "mm10" for mouse, "dre10" for zebrafish, "rn6" for rat, "dm6" for fruit fly, "ce11" for worm, "sc3" for yeast, "hg19" and "hg38" for human
`tad`	TAD genomic regions for the species. Predefined TAD regions or any new TAD regions can be used for the analysis. TAD regions must be formated as GRanges object. Predefined TAD regions are 'tad_hg19', 'tad_hg38', 'tad_mm10', 'tad_dmel' for hg19, hg38, mm9 and dm6 assembly, respectively.
`near`	Boolean value presents whether cis-neighbourhood should be considered in the analysis
`upstream`	Holds upstream distance from the transcription start position
`downstream`	Holds downstream distance from the transcription end position
`cellline`	Cell lines for TAD regions.

Value

List of protein coding genes that falls into the TAD regions

Examples

```
## Not run:
regions<-system.file("extdata", "ncRegion.txt", package = "NoRCE")
regionNC <- rtracklayer::import(regions, format = "BED")

r<-getTADOverlap(bedfile = regionNC,
                 tad = tad_hg19,
                 org_assembly = 'hg19',
                 cellline = 'HUVEC')

## End(Not run)
```

`getUCSC` *Get nearest genes for the window of the upstream/downstream region.*

Description

When downstream = 0 / upstream = 0, function converts bed formated regions to HUGO genes

Usage

```
getUCSC(
  bedfile,
  upstream,
  downstream,
  org_assembly = c("hg19", "hg38", "mm10", "dre10", "rn6", "dm6", "ce11", "sc3")
)
```

Arguments

`bedfile`	Bed formated input gene regions
`upstream`	Maximum upstream distance from the transcription start region of the input gene
`downstream`	Maximum downstream distance from the transcription end region of the input gene
`org_assembly`	genomee assembly of interest for the analysis. Possible assemblies are "mm10" for mouse, "dre10" for zebrafish, "rn6" for rat, "dm6" for fruit fly, "ce11" for worm, "sc3" for yeast, "hg19" and "hg38" for human

Value

genes

Examples

```
 ## Not run:
regions<-system.file("extdata", "ncRegion.txt", package = "NoRCE")
regionNC <- rtracklayer::import(regions, format = "BED")

neighbour <- getUCSC(bedfile = regionNC,
                     upstream = 1000,
                     downstream = 1000,
                     org_assembly = 'hg19')

## End(Not run)
```

`goEnrichment` *Perform enrichment analysis of the given genes*

Description

Perform enrichment analysis of the given genes

Usage

```
goEnrichment(
  genes,
  org_assembly = c("hg19", "hg38", "mm10", "dre10", "rn6", "dm6", "ce11", "sc3"),
  GOtype = c("BP", "CC", "MF"),
  pCut = 0.05,
  pAdjCut = 0.05,
  pAdjust = c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"),
  min = 5,
  backG = "",
  backGType = "pc_gene",
  enrichTest = c("hyper", "binom", "fisher", "chi")
)
```

Arguments

`genes`	Set of input genes. Supported format HUGO.
`org_assembly`	Genome assembly of interest for the analysis. Possible assemblies are "mm10" for mouse, "dre10" for zebrafish, "rn6" for rat, "dm6" for fruit fly, "ce11" for worm, "sc3" for yeast, "hg19" and "hg38" for human
`GOtype`	Hierarchical category of the GO ontology. Possible values are "BP"(default), "CC", "MF".
`pCut`	Threshold value for the pvalue. Default value is 0.05
`pAdjCut`	Cutoff value for the adjusted p-values using one of given method. Default value is 0.05.
`pAdjust`	Methods of the adjusted p-values. Possible methods are "bonferroni", "holm", "BH"(default)
`min`	Minimum number of gene that are required for enrichment. By default, it is set to 5
`backG`	The set of genes that tested against to the input (background gene)
`backGType`	Type of the background gene. If miRNA gene set is used for background gene, backGType should be set to the 'mirna'
`enrichTest`	Types of enrichment methods to perform enrichment analysis. Possible values are "hyper"(default), "binom", "fisher", "chi".

Value

GO enrichment results

Examples

```
## Not run:
subsetGene <- breastmRNA[1:30,]
breastEnr <- goEnrichment(genes = subsetGene,
                          org_assembly = 'hg19',
                          GOtype = 'MF',
                          min = 2)

## End(Not run)
```

`KeggEnrichment` *KEGG pathway enrichment*

Description

KEGG pathway enrichment

Usage

```
KeggEnrichment(
  genes,
  org_assembly = c("hg19", "hg38", "mm10", "dre10", "rn6", "dm6", "ce11", "sc3"),
  pCut = 0.05,
  pAdjCut = 0.05,
  pAdjust = c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"),
  min = 5,
  gmtFile = "",
  isSymbol = "",
  isGeneEnrich = ""
)
```

Arguments

`genes`	Input genes
`org_assembly`	Genome assembly of interest for the analysis. Possible assemblies are "mm10" for mouse, "dre10" for zebrafish, "rn6" for rat, "dm6" for fruit fly, "ce11" for worm, "sc3" for yeast, "hg19" and "hg38" for human
`pCut`	Threshold value for the pvalue. Default value is 0.05
`pAdjCut`	Cutoff value for the adjusted p-values using one of given method. Default value is 0.05.
`pAdjust`	Methods of the adjusted p-values. Possible methods are "holm", "hochberg", "hommel", "bonferroni", "BH", "BY","fdr", "none"
`min`	Minimum number of genes that are required for enrichment. By default, it is set to 5.
`gmtFile`	File path of the gmt file
`isSymbol`	Boolean value that controls the gene formats. If it is TRUE, gene format of the gmt file should be symbol. Otherwise, gene format must be ENTREZ ID.
`isGeneEnrich`	Boolean value whether gene enrichment should be performed

Value

KEGG pathway enrichment results

Examples

```
## Not run:
subsetGene <- breastmRNA[1:30,]

br_enr<-KeggEnrichment(genes = subsetGene,
                       org_assembly='hg19')

## End(Not run)
```

`listTAD` *List cell line of the given topological domain regions*

Description

List cell line of the given topological domain regions

Usage

```
listTAD(TADName)
```

Arguments

`TADName`	input TAD regions

Value

cell line of the input tad data

Examples

```
## Not run:
listTAD(TADName = tad_hg19)

## End(Not run)
```

`mirna` *Brain miRNA expression retrieved from the TCGA*

Description

Brain miRNA expression retrieved from the TCGA

Usage

```
mirna
```

Format

Not Available

Source

https://www.gencodegenes.org/

Examples

```
data(mirna)
```

`mirnaGOEnricher` *GO term enrichments of the microRNA genes with mRNAs that fall in the given upstream/downstream regions of the microRNA genes*

Description

GO term enrichments of the microRNA genes with mRNAs that fall in the given upstream/downstream regions of the microRNA genes

Usage

```
mirnaGOEnricher(
  gene,
 org_assembly = c("hg19", "hg38", "mm10", "dre10", "rn6", "dm6", "ce11", "sc3"),
  near = FALSE,
  target = FALSE,
  backGenes = "",
  backGType = "pc_gene",
  isTADSearch = FALSE,
  TAD = c(tad_hg19, tad_dmel, tad_hg38, tad_mm10),
  express = FALSE,
  isCustomExp = FALSE,
  cancer,
  exp1,
  exp2,
  label1 = "",
  label2 = "",
```

```
  isUnionCorGene = FALSE,
  databaseFile = ""
)
```

Arguments

`gene`	Input microRNA gene. It supports both pre-miRNA and mature miRNA, however, when target prediction is performed (target= TRUE), miRNA genes should be mature.
`org_assembly`	Genome assembly of interest for the analysis. Possible assemblies are "mm10" for mouse, "dre10" for zebrafish, "rn6" for rat, "dm6" for fruit fly, "ce11" for worm, "sc3" for yeast, "hg19" and "hg38" for human
`near`	Boolean value presents whether cis-neighbourhood should be considered in the analysis
`target`	Boolean value shows whether miRNA target prediction should be performed
`backGenes`	The set of genes that tested against to the input
`backGType`	Type of the background gene. If miRNA gene set is used for background gene, backGType should be set to the 'mirna'
`isTADSearch`	Boolean value that shows whether TAD analysis is performed. This value has to be TRUE for TAD analysis.
`TAD`	TAD genomic regions for the species. Predefined TAD regions or any new TAD regions can be used for the analysis. TAD regions must be formated as GRanges object. Predefined TAD regions are 'tad_hg19', 'tad_hg38', 'tad_mm10', 'tad_dmel' for hg19, hg38, mm9 and dm6 assembly, respectively.
`express`	Boolean variable whether co-expression analysis is performed. If this option is set to TRUE, co-expression analysis will be performed.
`isCustomExp`	Boolean variable whether co-expression analysis with custom data will be performed. When this option is set, exp1 and exp2 parameters must be defined.
`cancer`	Defines the name of the TCGA project code such as 'BRCA' for correlation analysis. Possible cancer types ACC, BLCA, BRCA, CESC, CHOL, COAD, COADREAD, DLBC, ESCA, GBMLGG, HNSC, KICH, KIPAN, KIRC, KIRP, LGG, LIHC, LUAD, LUSC, OV, PAAD, PCPG, PRAD, READ, SARC, SKCM, STAD, STES, TGCT, THCA, THYM, UCEC, UCS, UVM
`exp1`	Custom expression data matrix. Columns must be genes and rows must be patients. If gene names are provided as header, no need to redefine the headers(labels) of the expression data.
`exp2`	Custom expression data matrix. Columns must be genes and rows must be patients. If gene names are provided as header, no need to redefine the headers(labels) of the expression data.
`label1`	Gene names of the custom exp1 expression data. If it is not provided, column name of the exp1 data will be taken.
`label2`	Gene names of the custom exp2 expression data. If it is not provided, column name of the exp2 data will be taken.
`isUnionCorGene`	Boolean value that shows whether union of the output of the co-expression analysis and the other analysis should be considered
`databaseFile`	Path of miRcancer.db file

Value

MiRNA GO term enrichment object for the given input

Examples

```
## Not run:
subsetGene <- brain_mirna[1:30,]

miGO <-mirnaGOEnricher(gene=subsetGene,
                       org_assembly='hg19',
                       near = TRUE,
                       target = FALSE)
## End(Not run)
```

`mirnaPathwayEnricher` *Pathway enrichments of the microRNA genes with mRNAs that fall in the given upstream/downstream regions of the microRNA genes*

Description

Pathway enrichments of the microRNA genes with mRNAs that fall in the given upstream/downstream regions of the microRNA genes

Usage

```
mirnaPathwayEnricher(
  gene,
  org_assembly = c("hg19", "hg38", "mm10", "dre10", "rn6", "dm6", "ce11", "sc3"),
  near = FALSE,
  target = FALSE,
  isTADSearch = FALSE,
  TAD = c(tad_hg19, tad_dmel, tad_hg38, tad_mm10),
  gmtName = "",
  express = FALSE,
  isCustomExp = FALSE,
  cancer,
  exp1,
  exp2,
  label1 = "",
  label2 = "",
  isUnionCorGene = FALSE,
  databaseFile,
  isGeneEnrich = FALSE
)
```

Arguments

gene	Input microRNA gene. It supports both pre-miRNA and mature miRNA, however, when target prediction is performed(target= TRUE), miRNA genes should be mature.

`org_assembly`	Genome assembly of interest for the analysis. Possible assemblies are "mm10" for mouse, "dre10" for zebrafish, "rn6" for rat, "dm6" for fruit fly, "ce11" for worm, "sc3" for yeast, "hg19" and "hg38" for human
`near`	Boolean value presents whether cis-neighbourhood should be considered in the analysis
`target`	Boolean value shows whether miRNA target prediction should be performed
`isTADSearch`	Boolean value that shows whether TAD analysis is performed. This value has to be TRUE for TAD analysis.
`TAD`	TAD genomic regions for the species. Predefined TAD regions or any new TAD regions can be used for the analysis. TAD regions must be formated as GRanges object. Predefined TAD regions are 'tad_hg19', 'tad_hg38', 'tad_mm10', 'tad_dmel' for hg19, hg38, mm9 and dm6 assembly, respectively.
`gmtName`	Custom pathway gmt file
`express`	Boolean variable whether co-expression analysis is performed. If this option is set to TRUE, co-expression analysis will be performed.
`isCustomExp`	Boolean variable whether co-expression analysis with custom data will be performed. When this option is set, exp1 and exp2 parameters must be defined.
`cancer`	Defines the name of the TCGA project code such as 'BRCA' for correlation analysis. Possible cancer types ACC, BLCA, BRCA, CESC, CHOL, COAD, COADREAD, DLBC, ESCA, GBMLGG, HNSC, KICH, KIPAN, KIRC, KIRP, LGG, LIHC, LUAD, LUSC, OV, PAAD, PCPG, PRAD, READ, SARC, SKCM, STAD, STES, TGCT, THCA, THYM, UCEC, UCS, UVM
`exp1`	Custom expression data matrix. Columns must be genes and rows must be patients. If gene names are provided as header, no need to redefine the headers(labels) of the expression data.
`exp2`	Custom expression data matrix. Columns must be genes and rows must be patients. If gene names are provided as header, no need to redefine the headers(labels) of the expression data.
`label1`	Gene names of the custom exp1 expression data. If it is not provided, column name of the exp1 data will be taken.
`label2`	Gene names of the custom exp2 expression data. If it is not provided, column name of the exp2 data will be taken.
`isUnionCorGene`	Boolean value that shows whether union of the output of the co-expression analysis and the other analysis should be considered
`databaseFile`	Path of miRcancer.db file
`isGeneEnrich`	Boolean value whether gene enrichment should be performed

Value

MiRNA pathway enrichment object for the given input

Examples

```
## Not run:
miPath <- mirnaPathwayEnricher(gene = brain_mirna,
                               org_assembly = 'hg19',
                               near = TRUE)

## End(Not run)
```

`mirnaRegionGOEnricher` *GO enrichments of the microRNA regions with mRNAs that fall in the given upstream/downstream regions of the microRNA genes*

Description

GO enrichments of the microRNA regions with mRNAs that fall in the given upstream/downstream regions of the microRNA genes

Usage

```
mirnaRegionGOEnricher(
  region,
  org_assembly = c("hg19", "hg38", "mm10", "dre10", "rn6", "dm6", "ce11", "sc3"),
  near = FALSE,
  target = FALSE,
  backG = "",
  backGType = "pc-genes",
  isTADSearch = FALSE,
  TAD = c(tad_hg19, tad_dmel, tad_hg38, tad_mm10),
  express = FALSE,
  isCustomExp = FALSE,
  cancer,
  exp1,
  exp2,
  label1 = "",
  label2 = "",
  isUnionCorGene = FALSE,
  databaseFile
)
```

Arguments

`region`	MiRNA region in a bed format
`org_assembly`	Genome assembly of interest for the analysis. Possible assemblies are "mm10" for mouse, "dre10" for zebrafish, "rn6" for rat, "dm6" for fruit fly, "ce11" for worm, "sc3" for yeast, "hg19" and "hg38" for human
`near`	Boolean value presents whether cis-neighbourhood should be considered in the analysis
`target`	Boolean value shows whether miRNA target prediction should be performed
`backG`	The set of genes that tested against to the input
`backGType`	Type of the background gene. If miRNA gene set is used for background gene, backGType should be set to the 'mirna'
`isTADSearch`	Boolean value that shows whether TAD analysis is performed. This value has to be TRUE for TAD analysis.
`TAD`	TAD genomic regions for the species. Predefined TAD regions or any new TAD regions can be used for the analysis. TAD regions must be formated as GRanges object. Predefined TAD regions are 'tad_hg19', 'tad_hg38', 'tad_mm10', 'tad_dmel' for hg19, hg38, mm9 and dm6 assembly, respectively.

`express`	Boolean variable whether co-expression analysis is performed. If this option is set to TRUE, co-expression analysis will be performed.
`isCustomExp`	Boolean variable whether co-expression analysis with custom data will be performed. When this option is set, exp1 and exp2 parameters must be defined.
`cancer`	Defines the name of the TCGA project code such as 'BRCA' for correlation analysis. Possible cancer types ACC, BLCA, BRCA, CESC, CHOL, COAD, COADREAD, DLBC, ESCA, GBMLGG, HNSC, KICH, KIPAN, KIRC, KIRP, LGG, LIHC, LUAD, LUSC, OV, PAAD, PCPG, PRAD, READ, SARC, SKCM, STAD, STES, TGCT, THCA, THYM, UCEC, UCS, UVM
`exp1`	Custom expression data matrix. Columns must be genes and rows must be patients. If gene names are provided as header, no need to redefine the headers(labels) of the expression data.
`exp2`	Custom expression data matrix. Columns must be genes and rows must be patients. If gene names are provided as header, no need to redefine the headers(labels) of the expression data.
`label1`	Gene names of the custom exp1 expression data. If it is not provided, column name of the exp1 data will be taken.
`label2`	Gene names of the custom exp2 expression data. If it is not provided, column name of the exp2 data will be taken.
`isUnionCorGene`	Boolean value that shows whether union of the output of the co-expression analysis and the other analysis should be considered
`databaseFile`	Path of miRcancer.db file

Value

MiRNA GO enrichment object for the given input

Examples

```
## Not run:
regions<-system.file("extdata", "ncRegion.txt", package = "NoRCE")
regionNC <- rtracklayer::import(regions, format = "BED")

a<- mirnaRegionGOEnricher(region = regionNC,
                          org_assembly = 'hg19',
                          near = TRUE)

## End(Not run)
```

`mirnaRegionPathwayEnricher`

Pathway enrichments of the microRNA regions with mRNAs that fall in the given upstream/downstream regions of the microRNA genes

Description

Pathway enrichments of the microRNA regions with mRNAs that fall in the given upstream/downstream regions of the microRNA genes

Usage

```
mirnaRegionPathwayEnricher(
  region,
  org_assembly = c("hg19", "hg38", "mm10", "dre10", "rn6", "dm6", "ce11", "sc3"),
  near = FALSE,
  target = FALSE,
  isTADSearch = FALSE,
  TAD = c(tad_hg19, tad_dmel, tad_hg38, tad_mm10),
  gmtName = "",
  express = FALSE,
  isCustomExp = FALSE,
  cancer,
  exp1,
  exp2,
  label1 = "",
  label2 = "",
  isUnionCorGene = FALSE,
  databaseFile,
  isGeneEnrich = FALSE
)
```

Arguments

`region`	MiRNA region in a bed format
`org_assembly`	Genome assembly of interest for the analysis. Possible assemblies are "mm10" for mouse, "dre10" for zebrafish, "rn6" for rat, "dm6" for fruit fly, "ce11" for worm, "sc3" for yeast, "hg19" and "hg38" for human
`near`	Boolean value presents whether cis-neighbourhood should be considered in the analysis
`target`	Boolean value shows whether miRNA target prediction should be performed
`isTADSearch`	Boolean value that shows whether TAD analysis is performed. This value has to be TRUE for TAD analysis.
`TAD`	TAD genomic regions for the species. Predefined TAD regions or any new TAD regions can be used for the analysis. TAD regions must be formated as GRanges object. Predefined TAD regions are 'tad_hg19', 'tad_hg38', 'tad_mm10', 'tad_dmel' for hg19, hg38, mm9 and dm6 assembly, respectively.
`gmtName`	Custom pathway gmt file
`express`	Boolean variable whether co-expression analysis is performed. If this option is set to TRUE, co-expression analysis will be performed.
`isCustomExp`	Boolean variable whether co-expression analysis with custom data will be performed. When this option is set, exp1 and exp2 parameters must be defined.
`cancer`	Defines the name of the TCGA project code such as 'BRCA' for correlation analysis. Possible cancer types ACC, BLCA, BRCA, CESC, CHOL, COAD, COADREAD, DLBC, ESCA, GBMLGG, HNSC, KICH, KIPAN, KIRC, KIRP, LGG, LIHC, LUAD, LUSC, OV, PAAD, PCPG, PRAD, READ, SARC, SKCM, STAD, STES, TGCT, THCA, THYM, UCEC, UCS, UVM
`exp1`	Custom expression data matrix. Columns must be genes and rows must be patients. If gene names are provided as header, no need to redefine the headers(labels) of the expression data.

`exp2`	Custom expression data matrix. Columns must be genes and rows must be patients. If gene names are provided as header, no need to redefine the headers(labels) of the expression data.
`label1`	Gene names of the custom exp1 expression data. If it is not provided, column name of the exp1 data will be taken.
`label2`	Gene names of the custom exp2 expression data. If it is not provided, column name of the exp2 data will be taken.
`isUnionCorGene`	Boolean value that shows whether union of the output of the co-expression analysis and the other analysis should be considered
`databaseFile`	Path of miRcancer.db file
`isGeneEnrich`	Boolean value whether gene enrichment should be performed

Value

miRNA pathway enrichment object for the given input

Examples

```
## Not run:
regions<-system.file("extdata", "ncRegion.txt", package = "NoRCE")
regionNC <- rtracklayer::import(regions, format = "BED")

a<- mirnaRegionPathwayEnricher(region = regionNC,
             org_assembly = 'hg19')

## End(Not run)
```

`mrna` *Brain mRNA expression retrieved from the TCGA*

Description

Brain mRNA expression retrieved from the TCGA

Usage

```
mrna
```

Format

Not Available

Source

https://www.gencodegenes.org/

Examples

```
data(mrna)
```

`ncRegion` *Differentially expressed non-coding gene regions*

Description

Differentially expressed non-coding gene regions

Usage

```
ncRegion
```

Format

Not Available

Source

http://resource.psychencode.org/

Examples

```
data(ncRegion)
```

`NoRCE-class` *An S4 class to represent enrichment*

Description

An S4 class to represent enrichment

Slots

`ID` factor

`Term` factor

`geneList` factor

`ncGeneList` factor

`pvalue` factor

`pAdj` factor

`GeneRatio` factor

`BckRatio` factor

`packageCheck` *Check the package availability for the given assembly*

Description

Check the package availability for the given assembly

Usage

```
packageCheck(pkg)
```

Arguments

`pkg`	Required packages

Value

return install packages

`pathwayEnrichment` *For a given gmt file of a specific pathway database, pathway enrichment can be performed. Function supports Entrez ID and symbol based gmt file.*

Description

For a given gmt file of a specific pathway database, pathway enrichment can be performed. Function supports Entrez ID and symbol based gmt file.

Usage

```
pathwayEnrichment(
  genes,
  gmtFile,
  org_assembly = c("hg19", "hg38", "mm10", "dre10", "rn6", "dm6", "ce11", "sc3"),
  pCut = 0.05,
  pAdjCut = 0.05,
  pAdjust = c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"),
  isSymbol,
  min = 5,
  isGeneEnrich = FALSE
)
```

Arguments

`genes`	Input genes
`gmtFile`	File path of the gmt file
`org_assembly`	Genome assembly of interest for the analysis. Possible assemblies are "mm10" for mouse, "dre10" for zebrafish, "rn6" for rat, "dm6" for fruit fly, "ce11" for worm, "sc3" for yeast, "hg19" and "hg38" for human
`pCut`	Threshold value for the pvalue. Default value is 0.05
`pAdjCut`	Cutoff value for the adjusted p-values using one of given method. Default value is 0.05.
`pAdjust`	Methods of the adjusted p-values. Possible methods are "holm", "hochberg", "hommel", "bonferroni", "BH", "BY","fdr", "none"
`isSymbol`	Boolean value that controls the gene formats. If it is TRUE, gene format of the gmt file should be symbol. Otherwise, gene format must be ENTREZ ID.
`min`	Minimum number of genes that are required for enrichment. By default, it is set to 5.
`isGeneEnrich`	Boolean value whether gene enrichment should be performed

Value

Pathway Enrichment

`predictmiTargets` *Predict the miRNA targets for the miRNA or mRNA genes, which is specified with type parameter*

Description

Predict the miRNA targets for the miRNA or mRNA genes, which is specified with type parameter

Usage

```
predictmiTargets(gene, type, org_assembly)
```

Arguments

`gene`	Data frame of miRNA or mRNA gene. Formats should be NCBI gene name, ENSEMBL gene or transcript id, and mirna
`type`	Format of the gene, it should be "NCBI" for NCBI gene name, "Ensembl_gene" for ENSEMBL gene id, "Ensembl_trans" for Ensembl transcript id and "mirna" for miRNA gene
`org_assembly`	Analyzed genome assembly. Possible assemblies are "mm10" for mouse, "dre10" for zebrafish, "rn6" for rat, "dm6" for fruit fly, "ce11" for worm, "hg19" and "hg38" for human

Value

miRNA:mRNA target sets of the given genes

Examples

```
## Not run:
a<- predictmiTargets(gene = brain_mirna[1:100,],
                     org_assembly = 'hg19',
                     type = "mirna")

## End(Not run)
```

`reactomeEnrichment` *Reactome pathway enrichment*

Description

Reactome pathway enrichment

Usage

```
reactomeEnrichment(
  genes,
  org_assembly = c("hg19", "hg38", "mm10", "dre10", "rn6", "dm6", "ce11", "sc3"),
  pCut = 0.05,
  pAdjCut = 0.05,
  pAdjust = c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"),
  min = 5,
  gmtFile = "",
  isSymbol = "",
  isGeneEnrich = ""
)
```

Arguments

`genes`	Input genes
`org_assembly`	Genome assembly of interest for the analysis. Possible assemblies are "mm10" for mouse, "dre10" for zebrafish, "rn6" for rat, "dm6" for fruit fly, "ce11" for worm, "sc3" for yeast, "hg19" and "hg38" for human
`pCut`	Threshold value for the pvalue. Default value is 0.05
`pAdjCut`	Cutoff value for the adjusted p-values using one of given method. Default value is 0.05.
`pAdjust`	Methods of the adjusted p-values. Possible methods are "holm", "hochberg", "hommel", "bonferroni", "BH", "BY","fdr", "none"
`min`	Minimum number of genes that are required for enrichment. By default, it is set to 5.
`gmtFile`	File path of the gmt file
`isSymbol`	Boolean value that controls the gene formats. If it is TRUE, gene format of the gmt file should be symbol. Otherwise, gene format must be ENTREZ ID.
`isGeneEnrich`	Boolean value whether gene enrichment should be performed

Value

Reactome pathway enrichment results

Examples

```
## Not run:
br_enr<-reactomeEnrichment(genes = breastmRNA,org_assembly='hg19')
## End(Not run)
```

`setParameters` *Set the parameters*

Description

Parameters: upstream: Upstream distance from the transcription start position downstream: Downstream distance from the transcription end position searchRegion: Search space of the cis-region. Possible values are "all", "exon", "intron" GOtype: Hierarchical category of the GO ontology. Possible values are "BP", "CC", "MF" pCut: Threshold value for the pvalue. Default value is 0.05 pAdjCut: Cutoff value for the adjusted p-values using one of given method. Default value is 0.05. pAdjust: Methods of the adjusted p-values. Possible methods are "holm", "hochberg", "hommel", "bonferroni", "BH", "BY","fdr", "none" min: Minimum number of genes that are required for enrichment. By default, this value is set to 5. cellline: Cell lines for TAD regions. corrMethod Correlation coeffient method that will be used for evaluation. Possible values are "pearson", "kendall", "spearman" varCutoff: Variance cutt off that genes have less variance than this value will be trimmed pcut: P-value cut off for the correlation values alternate: Holds the alternative hypothesis and "two.sided", "greater" or "less" are the possible values. conf: Confidence level for the returned confidence interval. It is only used for the Pearson correlation coefficient if there are at least 4 complete pairs of observations. minAbsCor: Cut-off value for the Pearson correlation coefficient of the miRNA-mRNA pathwayType: Pathway database for enrichment. Possible values are 'reactome' for Reactome, 'kegg' for KEGG, 'wiki' for WikiPathways, 'other' for custom database enrichTest: Types of enrichment methods to perform enrichment analysis. Possible values are "hyper"(default), "binom", "fisher", "chi". isSymbol: Boolean variable that hold the gene format of the gmt file. If it is set as TRUE, gene format of the gmt file should be symbol. Otherwise, gene format should be ENTREZ ID. By default, it is FALSE.

Usage

```
setParameters(type, value)
```

Arguments

`type`	List of parameter names
`value`	New values for the parameters. Value and the parameter names must be in the same order.

Value

changed parameters

Examples

```
## Not run:
type <- c('downstream','upstream')

value <- c(2000,30000)

setParameters(type,value)

## End(Not run)
```

`tad_dmel` *TAD regions for the fly*

Description

TAD regions for the fly

Usage

```
tad_dmel
```

Format

Not Available

Source

http://chorogenome.ie-freiburg.mpg.de/data_sources.html#hi-c_datasets

Examples

```
data(tad_dmel)
```

`tad_hg19` *TAD regions for human hg19 assembly*

Description

TAD regions for human hg19 assembly

Usage

```
tad_hg19
```

Format

Not Available

Source

http://promoter.bx.psu.edu/hi-c/publications.html

Examples

```
data(tad_hg19)
```

`tad_hg38` *TAD regions for human hg38 assembly*

Description

TAD regions for human hg38 assembly

Usage

```
tad_hg38
```

Format

Not Available

Source

http://promoter.bx.psu.edu/hi-c/publications.html

Examples

```
data(tad_hg38)
```

`tad_mm10` *TAD regions for mouse*

Description

TAD regions for mouse

Usage

```
tad_mm10
```

Format

Not Available

Source

http://promoter.bx.psu.edu/hi-c/publications.html

Examples

```
data(tad_mm10)
```

`topEnrichment` *Number of top enrichment results of the pathway or GO terms for the given object and the order type - p-value or adjusted p-value.*

Description

Number of top enrichment results of the pathway or GO terms for the given object and the order type - p-value or adjusted p-value.

Usage

```
topEnrichment(mrnaObject, type, n)
```

Arguments

`mrnaObject`	Object of the enrichment result
`type`	Draw the dot plot according to the p-value or adjusted p-value ("pvalue", "pAdjust")
`n`	Number of GO terms or pathways, that ordered by type and has least number of top p-value

Value

Give top n enrichment results

Examples

```
## Not run:
ncGO<-geneGOEnricher(gene = brain_disorder_ncRNA, org_assembly='hg19',
   near=TRUE, genetype = 'Ensembl_gene')

result = topEnrichment(mrnaObject = ncGO, type = "pvalue", n = 10)

## End(Not run)
```

`WikiEnrichment` *WikiPathways Enrichment*

Description

WikiPathways Enrichment

Usage

```
WikiEnrichment(
  genes,
  org_assembly = c("hg19", "hg38", "mm10", "dre10", "rn6", "dm6", "ce11", "sc3"),
  pCut = 0.05,
  pAdjCut = 0.05,
  pAdjust = c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"),
  min = 5,
  gmtFile = "",
  isSymbol = "",
  isGeneEnrich = ""
)
```

Arguments

`genes`	Input genes
`org_assembly`	Genome assembly of interest for the analysis. Possible assemblies are "mm10" for mouse, "dre10" for zebrafish, "rn6" for rat, "dm6" for fruit fly, "ce11" for worm, "sc3" for yeast, "hg19" and "hg38" for human
`pCut`	Threshold value for the pvalue. Default value is 0.05
`pAdjCut`	Cutoff value for the adjusted p-values using one of given method. Default value is 0.05.
`pAdjust`	Methods of the adjusted p-values. Possible methods are "holm", "hochberg", "hommel", "bonferroni", "BH", "BY","fdr", "none"
`min`	Minimum number of genes that are required for enrichment. By default, it is set to 5.
`gmtFile`	File path of the gmt file
`isSymbol`	Boolean value that controls the gene formats. If it is TRUE, gene format of the gmt file should be symbol. Otherwise, gene format must be ENTREZ ID.
`isGeneEnrich`	Boolean value whether gene enrichment should be performed

Value

Wiki Pathway Enrichment

`writeEnrichment` *Write the tabular form of the pathway or GO term enrichment results*

Description

Write the tabular form of the pathway or GO term enrichment results

Usage

```
writeEnrichment(mrnaObject, fileName, sept = "\t", type = "pAdjust", n)
```

Arguments

`mrnaObject`	Object of the enrichment result
`fileName`	File name of the txt file
`sept`	File separator, by default, it is tab('\t')
`type`	Draw the dot plot according to the p-value or adjusted p-value ("pvalue", "pAdjust"). Default value is "pAdjust".
`n`	Number of GO terms or pathways, that ordered by type and has least number of top p-value

Value

Text file of the enrichment results in a tabular format

Examples

```
## Not run:
ncGO<-geneGOEnricher(gene = brain_disorder_ncRNA, org_assembly='hg19',
   near=TRUE, genetype = 'Ensembl_gene')

writeEnrichment(mrnaObject = ncGO,fileName = "a.txt",sept = '\t')

## End(Not run)
```

Index