fRNC

October 24, 2022

`case.exp_miRNA` *miRNA-seq expression data in the ECSA cancer samples*

Description

miRNA-seq expression data in the ECSA cancer samples, the row names are miRNA mirbase IDs, the column names are sample names in which last two identifier is 01

Usage

```
data("case.exp_miRNA")
```

Examples

```
data("case.exp_miRNA")
head(case.exp_miRNA[1:100,])
```

`case.exp_rna` *RNA-seq expression data in the ECSA cancer samples*

Description

RNA-seq expression data in the ECSA cancer samples, the row names are RNA mirbase IDs, the column names are sample names in which last two identifier is 01

Usage

```
data("case.exp_rna")
```

Examples

```
data("case.exp_rna")
head(case.exp_rna[1:100,])
```

combinp *Generate corrected p-value based on p-value and logFC in the expression matrix*

Description

based on the formula: corrected p-value = 2*(1-pnorm((-log10(p-value)) * abs(log2FC))) , corrected p-value was generated

Usage

```
combinp(node_attr = NULL, islog = T)
```

Arguments

node_attr	A data frame containing three columns: type, logFC and p value, and the row name is the gene identifier.
islog	Boolean value, whether to use the logFC, if FALSE, the weight is the p-value, or "TRUE", the corrected p-value is used

Value

A data matrix containing three columns: type, gene, weight(corrected p-value), the row name is the gene identifier.

References

Hongbo Shi, Jiayao Li, Qiong Song et al. (2019) Systematic identification and analysis of dysregulated miRNA and transcription factor feed-forward loops in hypertrophic cardiomyopathy

Examples

```
data("dataN")
result <- combinp(dataN[,c("type","logFC","PValue")])
```

control.exp_miRNA *miRNA-seq expression data in the ECSA normal samples*

Description

miRNA-seq expression data in the ECSA normal samples, the row names are miRNA mirbase IDs, the column names are sample names in which last two identifier is 11

Usage

```
data("control.exp_miRNA")
```

Examples

```
data("control.exp_miRNA")
head(control.exp_miRNA[1:100,])
```

control.exp_rna *RNA-seq expression data in the ECSA normal samples*

Description

RNA-seq expression data in the ECSA normal samples, the row names are RNA mirbase IDs, the column names are sample names in which last two identifier is 11

Usage

```
data("control.exp_rna")
```

Examples

```
data("control.exp_rna")
head(control.exp_rna[1:100,])
```

dataN *Example node_attr for ncRNA network*

Description

Data example consisting of a matrix of the colname of genes, type, logFC, logCPM, PValue, FDR.

Usage

```
data("dataN")
```

Examples

```
data("dataN")
head(dataN[1:100,])
```

DEGs *Performe differential expression analysis*

Description

Differential expression analysis using edgeR or limma for two group comparison

Usage

```
DEGs(case.exp, control.exp, geneid, data_type)
```

Arguments

case.exp	the case expression matrix, in which the row name is gene id and the column name is sample id
control.exp	the control expression matrix, in which the row name is gene id (the same with the case) and the column name is sample id
geneid	gene id in the case or control expression matrix
data_type	a character string indicating which date type to deal with is to be choosed, One of "RNAseq_counts" , "fpkm" and "microarray": can be abbreviated

Value

DEGlist contain four elements, DEGs the differential expression matrix, Nor_expr the normalized expression matrix, data_type a character string of the data type algorithm a character string indicating which algorithm was used, One of "edgeR" , "limma"

Examples

```
## Not run:
data("case.exp_miRNA")
data("control.exp_miRNA")
result_miR <- DEGs(case.exp_miRNA,control.exp_miRNA,
geneid= rownames(control.exp_miRNA), data_type = "RNAseq_counts")


## End(Not run)
```

ESCA_clinical *clinical data in the ECSA cancer patients*

Description

clinical data in the ECSA cancer patients, the column names are TCGA sample ID(cancer sample ID),survival time (month),survival status

Usage

```
data(ESCA_clinical)
```

Examples

```
data(ESCA_clinical)
head(ESCA_clinical[1:100,])
```

fRNC-package *fRNC: Search for modules in a node-weighted ncRNA network*

Description

fRNC constructs a node-weighted ncRNA network, performs module searching, generates simulation data from random networks, normalizes module scores using simulation data, removes unqualified modules, and orders resultant modules according to their significance.

Details

This package takes three types of data as input: a list of genes with association p-values and logFC, a human ncRNA network. generate_graph constructs a node-weighted ncRNA network. runmodule performs module search upon the node-weighted ncRNA network.

References

Hongbo Shi, Jiayao Li, Qiong Song et al. (2019) Systematic identification and analysis of dysregulated miRNA and transcription factor feed-forward loops in hypertrophic cardiomyopathy

Peilin Jia, Siyuan Zheng, Jirong Rong, Wei Zheng, Zhongming Zhao. (2011) Bioformatics. dmGWAS: dense module searching for genome-wide association studies in protein-protein interaction networks.

gene_type *Table of the gene identifier and the type for ensembl ID*

Description

Data consisting of a matrix, the colname contain gene_ID and gene_Name

Usage

```
data("gene_type")
```

Examples

```
data("gene_type")
head(gene_type[1:100,])
```

IDsymbol *Table of the gene identifier and the gene symbol*

Description

Data consisting of a matrix of the colname of genes, gene_Name and type

Usage

```
data("IDsymbol")
```

Examples

```
data("IDsymbol")
IDsymbol[1:100,]
```

integPvals *Integrate multiple p-values into joint p-values*

Description

The function integrate multiple p-values into the joint p-value of p-values based on the order statistics of p-values. An joint p-value #is given by the kth order statistic.

Usage

```
integPvals(pvalmatrix)
```

Arguments

pvalmatrix	Numeric matrix of p-values, columns represent different sets of p-values

Value

matrix The matrix of two columns: gene, weight(p-value)

`interStringency` *Extract interactions according to stringency and interaction type*

Description

interactions were extracted according to stringency and interaction type in the database of ENCORI

Usage

```
interStringency(
  type = c("Protein", "transcription"),
  spec = c("hg", "mm"),
  stringency = c("low", "medium", "high", "strict")
)
```

Arguments

`type`	a character string indicating which interaction type is to be choosed . One of "Protein" (RBP-circRNA, RBP-lncRNA, miRNA-circRNA, miRNA-lncRNA, RBP-miRNA, RBP-RBP), "transcription (miRNA-circRNA,miRNA-lncRNA,miRNA-RBP))": can be abbreviated
`spec`	a character string indicating which species is to be choosed . One of "hg", "mm": can be abbreviated
`stringency`	a character string indicating which interaction stringency is to be choosed . One of "low" (number of supported experiments > = 1 or combined_score >= 150 or score >=0), "medium (> = 2 or combined_score >= 400 or score >=5)","high (> = 3 or combined_score >= 700 or score >=10)","strict (> = 5 or combined_score >= 900, or score >=20)"

Value

Satisfactory interaction matrix. It contain five colnames:"node_gene_ID", "node_gene_Name", "type", "target_gene_ID", "target_gene_Name"

Examples

```
## Not run:
 interac <- interStringency(type = "Protein",spec ="hg", stringency = "strict")
 interac <- interac[,c("node_gene_ID","target_gene_ID")]

## End(Not run)
```

plotSub *Plot of the subnetwork*

Description

The function plots a network from graphNEL or igraph format. It is used to visualize the modules. For further plotting options use the plot.igraph function of the igraph package. The shapes of the nodes can be changed according to the scores argument, then negative scores appear squared The color of the nodes can be changed according to the diff.expr argument. Negative(positive) values lead to green(red) nodes.

Usage

```
plotSub(
  network,
  layout = layout.fruchterman.reingold,
  labels = NULL,
  diff.expr = NULL,
  scores = NULL,
  main = NULL,
  vertex.size = NULL
)
```

Arguments

network	A graph in igraph or graphNEL format.
layout	Layout algorithm, e.g. layout.fruchterman.reingold or layout.kamada.kawai.
labels	Labels for the nodes of the network
diff.expr	Named numerical vector of log2FC of the nodes in the network for coloring of the nodes.
scores	Named numerical vector of scores of the nodes for the shape of the node in the network.
main	Main title of the plot.
vertex.size	Numerical value or verctor for the size of the vertices.

References

Daniela Beisser, Gunnar W. Klau, Thomas Dandekar et al. (2010) BioNet: an R-Package for the functional analysis of biological networks

Examples

```
library(igraph)
edgel <- cbind(c("1", "2", "3", "4", "5", "6", "7"),
c("b", "c", "d", "e", "f", "a", "b"))
g <- graph.edgelist(edgel, directed=TRUE)
V(g)$type <- c(rep("lncRNA",4),rep("miRNA",4),rep("circRNA",5))
plotSub(g)
```

`runmodule` *Run module search function*

Description

runmodule constructs a node-weighted ncRNA network, performs module searching, generates simulation data from random networks, normalizes module scores using simulation data, removes unqualified modules, and orders resultant modules according to their significance.

Usage

```
runmodule(
  network,
  gene2weight,
  method = c("global", "local"),
  expr1 = NULL,
  expr2 = NULL,
  d = 2,
  r = 0.1,
  seletN = NULL,
  FDR = 1e-14,
  lambda = 0.5,
  min.size = 5,
  maxsize = 15,
  issymbol = TRUE
)
```

Arguments

`network`	A data frame containing a symbolic edge list of the ncRNA network in which the columns must contain "node_gene_ID", "type", "target_gene_Name"
`gene2weight`	A weigth data frame containing three columns:"type","gene", "weight" the first "type" the type of the gene identifier; lncRNA, miRNA, circRNA and RBP the second gene is unique, gene identifier (should be coordinate with the node symbol used in ncRNA network); the third weight is gene-based p-value or corrected p-value derived from differentially gene analysis or survival analysis
`method`	a character string indicating which the search method is to be computed . One of "global" (default, refer to Heinz method), "local (refer to GS method)": can be abbreviated
`expr1`	the expression matrix of the case sample
`expr2`	the expression matrix of the control sample
`d`	An integer used to define the order of neighbour genes to be searched in the method of the method "local" . This parameter is default set up as 2
`r`	A float indicating the cut-off for increment during module expanding process in the method of the method "local". Greater r will generate smaller module. Default is 0.1.
`seletN`	a vector: gene identifier IDs, or a gene identifier ID, for example "MIMAT0000461", or c("MIMAT0000461", "ENSG00000250742")

FDR	Numeric value, from the false discovery rate a p-value threshold is calculated. P-values below this threshold are considered to be significant The FDR can be used to control the size of the maximum scoring module
min.size	An integer: the min numbel of size of the module for user settings in the method of "global", default 5.
maxsize	An integer: the max numbel of size of the module for user settings in the method of "global", default 15.
issymbol	Boolean value, whether to set the node attribute "symbol"(gene symbol) in the network.

Value

runmodule returns a list containing relevant data and results, including:

GNCW	the node-weighted network used for searching
module	list of genes comprising each module, named for the seed gene if the method is "local" or t
module.score.matrix	contains Zm, Zn

References

Hongbo Shi, Jiayao Li, Qiong Song et al. (2019) Systematic identification and analysis of dysregulated miRNA and transcription factor feed-forward loops in hypertrophic cardiomyopathy

Peilin Jia, Siyuan Zheng, Jirong Rong, Wei Zheng, Zhongming Zhao. (2011) Bioformatics. dmGWAS: dense module searching for genome-wide association studies in protein-protein interaction networks.

Daniela Beisser, Gunnar W. Klau, Thomas Dandekar et al. (2019) BioNet: an R-Package for the functional analysis of biological networks

Examples

```
## Not run:
data("dataN")
gene2weight <- combinp(dataN[,c("type","logFC","PValue")])
interac <- interStringency(type = "transcription", spec ="hg",
stringency = "strict")
interac <- interac[,c("node_gene_ID","type","target_gene_ID")]
 res.list_global <- runmodule(network = interac, gene2weight,
  method = "global",FDR = 1e-14)
 res.list_local <- runmodule(network = interac, gene2weight,
  method = "local",maxsize=15, seletN = "MIMAT0000461")


## End(Not run)
```

---

savelocalM *save and plot module with local method*

Description

save and plot module for the method local result in the function runmodule

Usage

```
savelocalM(res.list_local)
```

Arguments

res.list_local	the methd local result

Value

the plot and the format XGMML of the each module, filenames is the seed node

Examples

```
## Not run:
data("dataN")
gene2weight <- combinp(dataN[,c("type","logFC","PValue")])
interac <- interStringency(type = "transcription", spec ="hg",
stringency = "strict")
interac <- interac[,c("node_gene_ID","type","target_gene_ID")]
res.list_local <- runmodule(network = interac, gene2weight,
 method = "local",maxsize=15, seletN = "MIMAT0000461")
savelocalM(res.list_local)

## End(Not run)
```

---

saveNetwork *save the subnetwork with global method*

Description

The function plots a RNC subnetwork from graphNEL or igraph format. It is used to visualize the modules. For further plotting options use the plot.igraph function of the igraph package. The shapes of the nodes can be changed according to the scores argument

Usage

```
saveNetwork(
  network,
  name = "network",
  file,
  type = c("table", "XGMML", "sif", "tab")
)
```

Arguments

network	A graph in igraph or graphNEL format.
name	Name of the network, only needed for the XGMML format.
file	File name to save.
type	Type in which graph shall be saved.

References

Daniela Beisser, Gunnar W. Klau, Thomas Dandekar et al. (2010) BioNet: an R-Package for the functional analysis of biological networks

Examples

```
## Not run:
data("dataN")
gene2weight <- combinp(dataN[,c("type","logFC","PValue")])
interac <- interStringency(type = "transcription", spec ="hg",
stringency = "strict")
interac <- interac[,c("node_gene_ID","type","target_gene_ID")]
res.list_global <- runmodule(network = interac, gene2weight,
method = "global",FDR = 1e-14)
saveNetwork(res.list_global$module,file="filenames",type = "XGMML")

## End(Not run)
```

`survival.km` *Performe univariate Cox regression analysis and plot Kaplan-Meier curve of the RNC module*

Description

univariate Cox regression analysis using survival with survival data and expression data,and Patients with higher and lower than the median risk score of the he dysregulated RNC are classified into different groups. Kaplan-Meier survival analysis was used to assess the clinical significance between the comparison groups

Usage

```
survival.km(
  gene_profile = NULL,
  clinData = NULL,
  genes = NULL,
  filename = "temp"
)
```

Arguments

`gene_profile`	the expression value matrix, in which the row name is gene id and the column name is sample id
`clinData`	the survival data, in which the column name is sample id, Survival(time) and Status(0,1)
`genes`	the gene in the RNC, The name of mature miRNA is miRbase ID, and name of lncRNA and RBP is Ensemble gene ID
`filename`	the output figure name in Kaplan-Meier curve

Value

contain the figure with the pdf format in Kaplan-Meier curve

`survival.rna` *Performe univariate Cox regression analysis*

Description

univariate Cox regression analysis using survival with survival data and expression data

Usage

```
survival.rna(gene_profile = NULL, clinData = NULL)
```

Arguments

`gene_profile`	the expression value matrix, in which the row name is gene id and the column name is sample id
`clinData`	the survival data, in which the column name is sample id, Survival(time) and Status(0,1)

Value

surivallist contain two elements, rna_p the survival result,in which the column name is gene id, P-value(from a univariable Cox proportional hazards regression model) and gene type(miRNA, lncRNA, RBP, circRNA) algorithm a character string indicating which algorithm was used with univariable Cox

Index