

Package ‘StarBioTrek’

April 12, 2018

Type Package

Title StarBioTrek

Version 1.4.0

Date 10-16-2017

Author Claudia Cava,
Isabella Castiglioni

Maintainer Claudia Cava <claudia.cava@ibfm.cnr.it>

Depends R (>= 3.3)

Imports SpidermiR, KEGGREST, org.Hs.eg.db, AnnotationDbi, e1071, ROCR,
grDevices, igraph

Description

This tool StarBioTrek presents some methodologies to measure pathway activity and cross-talk among pathways integrating also the information of network data.

License GPL (>= 3)

biocViews GeneRegulation, Network, Pathways, KEGG

Suggests BiocStyle, knitr, rmarkdown, testthat, devtools, roxygen2,
qgraph, png, grid

VignetteBuilder knitr

LazyData true

URL <https://github.com/claudiacava/StarBioTrek>

BugReports <https://github.com/claudiacava/StarBioTrek/issues>

RoxygenNote 6.0.1

NeedsCompilation no

R topics documented:

average	2
ds_score_crtlk	3
euc_dist_crtlk	3
getKEGGdata	4
getNETdata	4
GE_matrix	5
IPPI	5

list_path_net	6
matrix_plot	6
path_net	7
plotting_cross_talk	7
process_matrix	8
proc_path	8
SelectedSample	9
select_class	9
StarBioTrek	10
st_dv	10
svm_classification	11
Index	12

average	<i>For TCGA data get human pathway data and creates a matrix with the average of genes for each pathway.</i>
---------	--

Description

average creates a matrix with a summarized value for each pathway

Usage

```
average(dataFilt, pathway)
```

Arguments

dataFilt	TCGA matrix
pathway	pathway data

Value

a matrix value for each pathway

Examples

```
score_mean<-average(dataFilt=tumo[,1:2],path)
```

ds_score_crtlk	<i>For TCGA data get human pathway data and creates a measure of discriminating score among pathways</i>
----------------	--

Description

ds_score_crtlk creates a matrix with discriminating score for pathways

Usage

```
ds_score_crtlk(dataFilt, pathway)
```

Arguments

dataFilt	TCGA matrix
pathway	pathway data

Value

a matrix value for each pathway

Examples

```
cross_talk_st_dv<-ds_score_crtlk(dataFilt=tumo[,1:2],pathway=path)
```

euc_dist_crtlk	<i>For TCGA data get human pathway data and creates a measure of cross-talk among pathways</i>
----------------	--

Description

euc_dist_crtlk creates a matrix with euclidean distance for pairwise pathways

Usage

```
euc_dist_crtlk(dataFilt, pathway)
```

Arguments

dataFilt	TCGA matrix
pathway	pathway data

Value

a matrix value for each pathway

Examples

```
score_euc_dista<-euc_dist_crtlk(dataFilt=tumo[,1:2],path)
```

getKEGGdata *Get human KEGG pathway data.*

Description

getKEGGdata creates a data frame with human KEGG pathway. Columns are the pathways and rows the genes inside those pathway

Usage

```
getKEGGdata(KEGG_path)
```

Arguments

KEGG_path variable

Value

dataframe with human pathway data

Examples

```
path<-getKEGGdata(KEGG_path="Transcript")
```

getNETdata *Get network data.*

Description

getNETdata creates a data frame with network data. Network category can be filtered among: physical interactions, co-localization, genetic interactions and shared protein domain.

Usage

```
getNETdata(network, organism = NULL)
```

Arguments

network variable. The user can use the following parameters based on the network types to be used. PHint for Physical_interactions, COloc for Co-localization, GENint for Genetic_interactions and SHpd for Shared_protein_domains

organism organism==NULL default value is homo sapiens

Value

dataframe with gene-gene (or protein-protein interactions)

Examples

```
organism="Saccharomyces_cerevisiae"
netw<-getNETdata(network="SHpd",organism)
```

GE_matrix	<i>Get human KEGG pathway data and a gene expression matrix in order to obtain a matrix with the gene expression for only pathways given in input .</i>
-----------	---

Description

GE_matrix creates a matrix of gene expression for pathways given by the user.

Usage

```
GE_matrix(DataMatrix, pathway)
```

Arguments

DataMatrix	gene expression matrix (eg.TCGA data)
pathway	pathway data as provided by getKEGGdata

Value

a matrix for each pathway (gene expression level belong to that pathway)

Examples

```
list_path_gene<-GE_matrix(DataMatrix=tumo[,1:2],pathway=path)
```

IPPI

Multilayer analysis Cava et al. BMC Genomics 2017

Description

IPPI function takes as input pathway and network data in order to select genes with central role in that pathway. Please see Cava et al. 2017 BMC Genomics

Usage

```
IPPI(patha, netwa)
```

Arguments

patha	pathway matrix Please see example path for format
netwa	a dataframe Please see example path for format netw

Value

a list with driver genes for each pathway

Examples

```
DRIVER_SP<-IPPI(patha=path,netwa=netw)
```

list_path_net	<i>Get human KEGG pathway data and output of path_net in order to define the common genes.</i>
---------------	--

Description

list_path_net creates a list of interacting genes for each human pathway.

Usage

```
list_path_net(lista_net, pathway)
```

Arguments

lista_net	output of path_net
pathway	pathway data as provided by getKEGGdata

Value

a list of genes for each pathway (interacting genes belong to that pathway)

Examples

```
lista_netw<-path_net(pathway=path,data=netw)
list_path<-list_path_net(lista_net=lista_netw,pathway=path)
```

matrix_plot	<i>Get human KEGG pathway data and a gene expression matrix in order to obtain a matrix with the mean gene expression for only pathways given in input .</i>
-------------	--

Description

GE_matrix creates a matrix of mean gene expression for pathways given by the user.

Usage

```
matrix_plot(DataMatrix, pathway)
```

Arguments

DataMatrix	gene expression matrix (eg.TCGA data)
pathway	pathway data as provided by getKEGGdata

Value

a matrix for each pathway (mean gene expression level belong to that pathway)

Examples

```
list_path_plot<-matrix_plot(DataMatrix=tumo[,1:2],pathway=path)
```

path_net	<i>Get human KEGG pathway data and network data in order to define the common gene.</i>
----------	---

Description

path_net creates a list of network data for each human pathway. The network data will be generated when interacting genes belong to that pathway.

Usage

```
path_net(pathway, data)
```

Arguments

pathway	pathway data as provided by getKEGGdata
data	network data as provided by getNETdata

Value

a list of network data for each pathway (interacting genes belong to that pathway)

Examples

```
lista_net<-path_net(pathway=path,data=netw)
```

plotting_cross_talk	<i>Get human KEGG pathway data and a gene expression matrix we obtain a matrix with the gene expression for only pathways given in input</i>
---------------------	--

Description

plotting_matrix creates a matrix of gene expression for pathways given by the user.

Usage

```
plotting_cross_talk(DataMatrix, pathway, path_matrix)
```

Arguments

DataMatrix	gene expression matrix (eg.TCGA data)
pathway	pathway data as provided by getKEGGdata
path_matrix	output of the function matrix_plot

Value

a plot for pathway cross talk

Examples

```
mt<-plotting_cross_talk(DataMatrix=tumo[,1:2],pathway=path,path_matrix=list_path_plot)
```

process_matrix	<i>Process matrix TCGA data after the selection of pairwise pathway</i>
----------------	---

Description

processing gene expression matrix

Usage

```
process_matrix(measure, list_perf)
```

Arguments

measure	matrix with measure of cross-talk among pathways
list_perf	output of the function select_class

Value

a gene expression matrix for case study 1

proc_path	<i>Get human KEGG pathway data.</i>
-----------	-------------------------------------

Description

getKEGGdata creates a data frame with human KEGG pathway. Columns are the pathways and rows the genes inside those pathway

Usage

```
proc_path(mer)
```

Arguments

mer	output for example of select_path_carb
-----	--

Value

dataframe with human pathway data

SelectedSample	<i>Select the class of TCGA data</i>
----------------	--------------------------------------

Description

select two labels from ID barcode

Usage

```
SelectedSample(Dataset, typesample)
```

Arguments

Dataset	gene expression matrix
typesample	the labels of the samples (e.g. tumor,normal)

Value

a gene expression matrix of the samples with specified label

Examples

```
tumo<-SelectedSample(Dataset=Data_CANCER_normUQ_filt,typesample="tumor")[,2]
```

select_class	<i>Select the class of TCGA data</i>
--------------	--------------------------------------

Description

select two labels from ID barcode

Usage

```
select_class(auc.df, cutoff)
```

Arguments

auc.df	list of AUC value
cutoff	cut-off for AUC value

Value

a gene expression matrix with only pairwise pathway with a particular cut-off

StarBioTrek	<i>Download data</i>
-------------	----------------------

Description

StarBioTrek allows you to Download data of samples from StarBioTrek

Details

The functions you're likely to need from **StarBioTrek** is `path_star` Otherwise refer to the vignettes to see how to format the documentation.

st_dv	<i>For TCGA data get human pathway data and creates a measure of standard deviations among pathways</i>
-------	---

Description

st_dv creates a matrix with standard deviation for pathways

Usage

```
st_dv(DataMatrix, pathway)
```

Arguments

DataMatrix	TCGA matrix
pathway	pathway data

Value

a matrix value for each pathway

Examples

```
stand_dev<-st_dv(DataMatrix=tumo[,1:2],pathway=path)
```

svm_classification *SVM classification for each feature*

Description

svm class creates a list with auc value

Usage

```
svm_classification(TCGA_matrix, tumour, normal, nfs)
```

Arguments

TCGA_matrix	gene expression matrix
tumour	barcode samples for a class
normal	barcode samples for another class
nfs	nfs split data into a training and test set

Value

a list with AUC value for pairwise pathway

Examples

```
nf <- 60  
res_class <- svm_classification(TCGA_matrix=score_euc_dist, nfs=nf,  
normal=colnames(norm[,1:10]), tumour=colnames(tumo[,1:10]))
```

Index

average, [2](#)
ds_score_crtlk, [3](#)
euc_dist_crtlk, [3](#)
GE_matrix, [5](#)
getKEGGdata, [4](#)
getNETdata, [4](#)
IPPI, [5](#)
list_path_net, [6](#)
matrix_plot, [6](#)
path_net, [7](#)
plotting_cross_talk, [7](#)
proc_path, [8](#)
process_matrix, [8](#)
select_class, [9](#)
SelectedSample, [9](#)
st_dv, [10](#)
StarBioTrek, [10](#)
StarBioTrek-package (StarBioTrek), [10](#)
svm_classification, [11](#)