Package 'PMAPscore'

January 20, 2025

Type Package

Title Identify Prognosis-Related Pathways Altered by Somatic Mutation

Version 0.1.1

Maintainer Junwei Han <hanjunwei1981@163.com>

Description We innovatively defined a pathway mutation accumulate perturbation score (PMAPscore) to reflect the position and the cumulative effect of the genetic mutations at the pathway level. Based on the PMAPscore of pathways, identified prognosis-related pathways altered by somatic mutation and predict immunotherapy efficacy by constructing a multiple-pathway-based risk model (Tarca, Adi Laurentiu et al (2008) <doi:10.1093/bioinformatics/btn577>).

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

Imports base, clusterProfiler, glmnet, graphics, grDevices, maftools, org.Hs.eg.db, pROC, stats, survival, survminer, utils

VignetteBuilder knitr

Depends R (>= 4.1.0)

Suggests knitr, rmarkdown

NeedsCompilation no

Author Junwei Han [aut, cre, cph], Yalan He [aut], Xiangmei Li [aut]

Repository CRAN

Date/Publication 2022-04-12 08:12:37 UTC

Contents

`final_signature` *final_signature, the final prognosis-related pathways*

Description

The final prognosis-related pathways identified by our approach.

Usage

```
final_signature
```

Format

An object of class `character` of length 7.

`gene_symbol_Entrez` *gene_symbol_Entrez, the genes' symbol and ENTREZID*

Description

The genes' symbol and ENTREZID.

Usage

```
gene_symbol_Entrez
```

Format

An object of class `data.frame` with 54245 rows and 2 columns.

`gene_Ucox` *gene_Ucox*

Description

gene_Ucox

Usage

```
gene_Ucox
```

Format

An object of class `data.frame` with 4287 rows and 5 columns.

`gene_Ucox_res` *gene_Ucox_res, the univariate Cox regression result of candidate genes.*

Description

The univariate Cox regression result of candidate genes.

Usage

```
gene_Ucox_res
```

Format

An object of class `data.frame` with 4287 rows and 5 columns.

get_Entrez_ID *Convert gene symbol to Entrez_Gene_ID*

Description

The function 'get_Entrez_ID' is used to convert gene symbol to Entrez_Gene_ID

Usage

```
get_Entrez_ID(mut_status, gene_symbol_Entrez, Entrez_ID = TRUE)
```

Arguments

mut_status	A binary matrix that contains the mutation state of genes in each sample and its row name is the gene symbol. Noted the matrix can be generated by the function 'get_mut_status'.
gene_symbol_Entrez	A data table containing gene symbol and the corresponding gene Entrez ID.
Entrez_ID	Logical,tell whether there are Entrez IDs corresponding to gene symbol in the gene_symbol_Entrez.

Value

A binary matrix that contains the mutation state of genes in each sample and its row name is Entrez_Gene_ID.

Examples

```
#load the data.
data(mut_status,gene_symbol_Entrez)
#perform function `get_Entrez_ID`.
mut_status<-get_Entrez_ID(mut_status,gene_symbol_Entrez,Entrez_ID=TRUE)
```

get_final_signature *Identify the candidate prognosis-related pathways*

Description

The function 'get_final_signature' uses to identify the candidate prognosis-related pathways based on the PMAPscore.

Usage

```
get_final_signature(pfs_score, sur, wilcox_p = 0.05, uni_cox_p = 0.01)
```

Arguments

pfs_score	A 2 X n matrix that contains the pfs_score in each sample of the signal pathways. Noted the matrix can be generated by the function 'get_pfs_score'.
sur	This data contains survival status and survival time of each sample.
wilcox_p	The threshold of p value for Wilcoxon rank-sum test.
uni_cox_p	The threshold of p value for univariate Cox regression analysis.

Value

Return the candidate prognosis-related pathways

Examples

```
#load the data.
data(pfs_score,sur)
#perform function `get_final_signature`.
final_signature<-get_final_signature(pfs_score,sur)
```

---

get_km_survival_curve *Plot Kaplan-Meier survival curve.*

Description

The function 'get_km_survival_curve' uses to draw the Kaplan-Meier survival curve.

Usage

```
get_km_survival_curve(km_data, cut_point, TRAIN = TRUE, risk.table = TRUE)
```

Arguments

km_data	A data frame, including survival status, survival time, and risk score of each sample. The data frame can be generated by the function 'get_risk_score'.
cut_point	The threshold uses to classify patients into two subgroups with different OS.
TRAIN	Logical,if set to TRUE,the 'cut_point' is generated by the median of the risk score; Otherwise,'cut_point' can be customized.
risk.table	Allowed values include:TRUE or FALSE specifying whether to show or not the risk table. Default is FALSE.

Value

No return, plot the Kaplan-Meier survival curve.

Examples

```
#load the data.
data(km_data)
#perform the function `get_km_survival_curve`.
get_km_survival_curve(km_data,cut_point,TRAIN = TRUE,risk.table=TRUE)
```

`get_MultivariateCox_result` *Perform the multivariate Cox regression*

Description

The function 'get_MultivariateCox_result' uses to perform multivariate Cox regression analysis on the cancer-specific dysregulated signaling pathways.

Usage

```
get_MultivariateCox_result(DE_path_sur)
```

Arguments

`DE_path_sur`	A binary metadata table containing sample survival status and survival time.Note that the column names of survival time and survival status must be "survival" and "event".

Value

Return the multivariate Cox regression results of cancer-specific dysregulated signaling pathways.

Examples

```
#Load the data.
data(path_cox_data)
#perform function `get_MultivariateCox_result`.
res<-get_MultivariateCox_result(path_cox_data)
```

get_mut_status *Converts MAF file into mutation matrix*

Description

The function 'get_mut_status' uses to convert MAF file into mutation matrix.

Usage

```
get_mut_status(maf_data, nonsynonymous = TRUE)
```

Arguments

maf_data	The patients' somatic mutation data, which in MAF format.
nonsynonymous	Logical, tell if extract the non-silent somatic mutations (nonsense mutation, missense mutation, frame-shif indels, splice site, nonstop mutation, translation start site, inframe indels).

Value

A binary mutations matrix, in which 1 represents that a particular gene has mutated in a particular sample, and 0 represents that gene has no mutation in a particular sample .

Examples

```
#load the data
data(maf_data)
#perform the function `get_mut_status`.
mutmatrix.example<-get_mut_status(maf_data,nonsynonymous = TRUE)
```

get_Oncoplots *draw an GenePathwayOncoplots*

Description

Load the data in MAF format and draws an GenePathwayOncoplots.

Usage

```
get_Oncoplots(
  maffile,
  path_gene,
  mut_status,
  risk_score,
  cut_off,
  final_signature,
```

```
  pathway_name,
  isTCGA = FALSE,
  top = 20,
  clinicalFeatures = "sample_group",
  annotationColor = c("red", "green"),
  sortByAnnotation = TRUE,
  removeNonMutated = FALSE,
  drawRowBar = TRUE,
  drawColBar = TRUE,
  leftBarData = NULL,
  leftBarLims = NULL,
  rightBarData = NULL,
  rightBarLims = NULL,
  topBarData = NULL,
  logColBar = FALSE,
  draw_titv = FALSE,
  showTumorSampleBarcodes = FALSE,
  fill = TRUE,
  showTitle = TRUE,
  titleText = NULL
)
```

Arguments

maffile	A data of MAF format.
path_gene	User input pathways geneset list.
mut_status	The mutations matrix,generated by ‘get_mut_matrix‘.
risk_score	Samples’ PTMB-related risk score,which could be a biomarker for survival analysis and immunotherapy prediction.
cut_off	A threshold value(the median risk score as the default value).Using this value to divide the sample into high and low risk groups with different overall survival.
final_signature	The pathway signature,use to map gene in the GenePathwayOncoplots.
pathway_name	The name of the pathway that you want to visualize.For example "Gap junction"
isTCGA	Is input MAF file from TCGA source. If TRUE uses only first 12 characters from Tumor_Sample_Barcode.
top	How many top genes to be drawn,genes are arranged from high to low depending on the frequency of mutations. defaults to 20.
clinicalFeatures	Columns names from ’clinical.data’ slot of MAF to be drawn in the plot. Dafault "sample_group".
annotationColor	Custom colors to use for sample annotation-"sample_group". Must be a named list containing a named vector of colors. Default "red" and "green".
sortByAnnotation	Logical sort oncomatrix (samples) by provided ’clinicalFeatures’. Sorts based on first ’clinicalFeatures’. Defaults to TRUE. column-sort.

`removeNonMutated`
: Logical. If TRUE removes samples with no mutations in the GenePathwayOncoplots for better visualization. Default FALSE.

`drawRowBar`
: Logical. Plots righ barplot for each gene. Default TRUE.

`drawColBar`
: Logical plots top barplot for each sample. Default TRUE.

`leftBarData`
: Data for leftside barplot. Must be a data.frame with two columns containing gene names and values. Default 'NULL'.

`leftBarLims`
: Limits for 'leftBarData'. Default 'NULL'.

`rightBarData`
: Data for rightside barplot. Must be a data.frame with two columns containing to gene names and values. Default 'NULL' which draws distibution by variant classification. This option is applicable when only 'drawRowBar' is TRUE.

`rightBarLims`
: Limits for 'rightBarData'. Default 'NULL'.

`topBarData`
: Default 'NULL' which draws absolute number of mutation load for each sample. Can be overridden by choosing one clinical indicator(Numeric) or by providing a two column data.frame contaning sample names and values for each sample. This option is applicable when only 'drawColBar' is TRUE.

`logColBar`
: Plot top bar plot on log10 scale. Default FALSE.

`draw_titv`
: Logical Includes TiTv plot. Default FALSE

`showTumorSampleBarcodes`
: Logical to include sample names.

`fill`
: Logical. If TRUE draws genes and samples as blank grids even when they are not altered.

`showTitle`
: Default TRUE.

`titleText`
: Custom title. Default 'NULL'.

Value

No return value

Examples

```
#obtain the risksciore
data(km_data)
risk_score<-km_data$multiple_score
names(risk_score)<-rownames(km_data)
cut_off<-median(risk_score)
#load the dtata
data(final_signature,path_gene,mut_status,maffile)
##draw an GenePathwayOncoplots
get_Oncoplots(maffile,path_gene,mut_status,risk_score,cut_off,final_signature,"Gap junction")
```

get_pfs_score *Calculates the pathway-based mutation accumulate perturbation score*

Description

The function ‘get_pfs_score‘ uses to calculate the pathway-based mutation accumulate perturbation score using the matrix of gene mutation state and pathway information.

Usage

```
get_pfs_score(
  mut_status,
  percent,
  gene_Ucox_res,
  gene_symbol_Entrez,
  data.dir = NULL,
  organism = "hsa",
  verbose = TRUE,
  Entrez_ID = TRUE,
  gene_set = NULL
)
```

Arguments

mut_status	Mutation status of a particular gene in a particular sample. The file can be generated by the function ‘get_mut_status‘.
percent	This parameter is used to control the mutation rate of gene. Genes less than this value will be deleted
gene_Ucox_res	Results of gene univariate Cox regression.
gene_symbol_Entrez	A data table containing gene symbol and gene Entrez ID.
data.dir	Location of the "organism"SPIA.RData file containing the pathways data.If set to NULL will look for this file in the extdata folder of the PFS library.
organism	A three letter character designating the organism. See a full list at ftp://ftp.genome.jp/pub/kegg/xml/organi
verbose	If set to TRUE, displays the number of pathways already analyzed.
Entrez_ID	Logical,tell whether there are Entrez IDs corresponding to gene symbol in the gene_symbol_Entrez.
gene_set	A group of cancer specific gene symbols obtained from the training set

Value

A binary mutations matrix, which column names is sample and the row name is the pathway.

Examples

```
#get the path of the mutation annotation file.
data(mut_status,gene_Ucox_res,gene_symbol_Entrez)
#perform the function `get_pfs_score`.
pfs_score<-get_pfs_score(mut_status[,1:2],percent=0.03,gene_Ucox_res,gene_symbol_Entrez)
```

`get_response_plot` *Plot the response column diagram*

Description

The function 'get_response_plot' uses to plot the column diagram of drug response.

Usage

```
get_response_plot(km_data, response, cut_point, TRAIN = TRUE)
```

Arguments

`km_data`	A data frame, including survival status, survival time, and risk score of each sample. The data frame can be generated by the function 'get_risk_score'.
`response`	Response status of the sample to the drug.
`cut_point`	The threshold uses to classify patients into two subgroups with different OS.
`TRAIN`	Logical,if set to TRUE,the 'cut_point' is generated by the median of the risk score; Otherwise,'cut_point' can be customized.

Value

Comparison of the objective response rate between the high-risk and low-risk groups, plot the bar graph and return the p value.

Examples

```
#Load the data.
data(km_data,response)
#perform the function `get_response_plot`.
get_response_plot(km_data,response,cut_point,TRAIN=TRUE)
```

get_risk_score *Calculates the risk score for patients*

Description

The function 'get_risk_score' uses to calculate the risk score for patients based on cancer-specific dysregulated signaling pathways.

Usage

```
get_risk_score(
  final_signature,
  pfs_score,
  path_Ucox_mul_res,
  sur,
  TRAIN = TRUE
)
```

Arguments

final_signature	Cancer-specific dysregulated signal pathways. It can be generated by the function 'get_final_signature'.
pfs_score	A matrix that contains the pfs_score in each sample of the signal pathways. Noted the matrix can be generated by the function 'get_pfs_score'.
path_Ucox_mul_res	Results of multivariate Cox regression of cancer specific pathway in training set.
sur	This data contains survival status and survival time of each sample.
TRAIN	Logical,if set FLASE,we need to load the result of multivariate Cox regression of cancer specific pathways into the training set.

Value

A data set with the risk score for each sample.

Examples

```
#Load the data.
data(final_signature,pfs_score,sur,path_Ucox_mul_res)
#perform the function `get_risk_score`.
km_data<-get_risk_score(final_signature,pfs_score,path_Ucox_mul_res,sur,TRAIN=TRUE)
```

`get_roc_curve` *Plot the ROC curve*

Description

The function 'get_roc_curve' uses to plot the ROC curve for predicting immunotherapy response.

Usage

```
get_roc_curve(roc_data, print.auc = TRUE, main = "Objective Response")
```

Arguments

`roc_data`	A 2 X n data fram, which contain the immunotherapy response and risk score (generated by the function 'get_risk_score') for patients.
`print.auc`	Boolean. Should the numeric value of AUC be printed on the plot?
`main`	A main title for the plot.

Value

No return, plot the ROC curve for immunotherapy response prediction.

Examples

```
#Load the data.
data(roc_data)
#perform the function `get_roc_curve`.
get_roc_curve(roc_data,print.auc=TRUE,main="Objective Response")
```

`get_sam_cla` *get_sam_cla*

Description

Function 'get_sample_classification' This function is used to judge the classification of samples.

Usage

```
get_sam_cla(
  mut_sam,
  gene_Ucox,
  symbol_Entrez,
  path_cox_data,
  sur,
  path_Ucox_mul,
```

```
  sig,
  cut_off = -0.986,
  data.dir = NULL,
  organism = "hsa",
  TRAIN = FALSE
)
```

Arguments

mut_sam	The sample somatic mutation data.
gene_Ucox	Results of gene univariate Cox regression.
symbol_Entrez	A data table containing gene symbol and gene Entrez ID.
path_cox_data	Pathways of Cancer-specifical obtained from the training set.
sur	This data contains survival status and survival time of each sample.
path_Ucox_mul	Multivariate Cox regression results of Cancer-specifical pathways.
sig	Cancer-specific dysregulated signal pathways. It can be generated by the function ‘get_final_signature‘.
cut_off	Threshold of classification.
data.dir	Location of the "organism"SPIA.RData file containing the pathways data. If set to NULL will look for this file in the extdata folder of the PMAPscore library.
organism	A three letter character designating the organism. See a full list at ftp://ftp.genome.jp/pub/kegg/xml/organi
TRAIN	Logical,if set FLASE,we need to load the result of multivariate Cox regression of cancer specific pathways into the training set.

Value

Return a data frame, the sample’s risk score and the sample’s risk group.

Examples

```
#Load the data.
data(mut_sam,gene_Ucox,symbol_Entrez,path_cox_data,sur,path_Ucox_mul)
#perform function `get_sample_cla`.
get_sam_cla(mut_sam,gene_Ucox,symbol_Entrez,path_cox_data,sur,path_Ucox_mul,sig,cut_off=-0.986)
```

---

get_univarCox_result *Perform the univariate Cox regression analysis.*

Description

The function ‘get_univarCox_result‘ uses to perform the univariate Cox regression analysis.

Usage

```
get_univarCox_result(DE_path_sur)
```

Arguments

DE_path_sur	A binary metadata table containing survival status and survival time of each sample.Note that the column names of survival time and survival status must be "survival" and "event"

Value

Return a data frame, the univariate Cox regression analysis results.

Examples

```
#get path of the mutation annotation file.
data(path_cox_data)
#perform function `get_univarCox_result`.
res<-get_univarCox_result(path_cox_data)
```

---

km_data *km_data*

Description

The data use for drawing K-M survival curve.

Usage

```
km_data
```

Format

An object of class `data.frame` with 105 rows and 10 columns.

---

maffile *maffile*

Description

The mutation data of patients.

Usage

```
maffile
```

Format

An object of class `MAF` of length 1.

maf_data *maf_data*

Description

The mutation data of patients.

Usage

```
maf_data
```

Format

An object of class `data.frame` with 24461 rows and 4 columns.

mut_num *mut_num*

Description

mut_num

Usage

```
mut_num
```

Format

An object of class `matrix` (inherits from `array`) with 13858 rows and 105 columns.

mut_sam *mut_sam*

Description

mut_sam.

Usage

```
mut_sam
```

Format

An object of class `matrix` (inherits from `array`) with 13858 rows and 2 columns.

mut_sample *mut_sample*

Description

mut_sample.

Usage

```
mut_sample
```

Format

An object of class `matrix` (inherits from `array`) with 13858 rows and 2 columns.

mut_status *mut_status*

Description

mut_status.

Usage

```
mut_status
```

Format

An object of class `matrix` (inherits from `array`) with 13858 rows and 105 columns.

newspia *newspia*

Description

Function ‘newspia‘ This function is based on SPIA algorithm to analyse KEGG signal pathway for single sample..

Usage

```
newspia(
  de = NULL,
  all = NULL,
  organism = "hsa",
  data.dir = NULL,
  pathids = NULL,
  verbose = TRUE,
  beta = NULL
)
```

Arguments

de	A named vector containing the statue of particular genes in a particular sample.The names of this numeric vector are Entrez gene IDs.
all	A vector with the Entrez IDs in the reference set. If the data was obtained from a microarray experiment,this set will contain all genes present on the specific array used for the experiment.This vector should contain all names of the de argument.
organism	A three letter character designating the organism. See a full list at ftp://ftp.genome.jp/pub/kegg/xml/organi
data.dir	Location of the "organism"SPIA.RData file containing the pathways data .If set to NULL will look for this file in the extdata folder of the PMAPscore library.
pathids	A character vector with the names of the pathways to be analyzed.If left NULL all pathways available will be tested.
verbose	If set to TRUE, displays the number of pathways already analyzed.
beta	Weights to be assigned to each type of gene/protein relation type. It should be a named numeric vector of length 23, whose names must be: c("activation","compound","binding/associatio "inhibition","activation_phosphorylation","phosphorylation", "indirect","inhibition_phosphorylation","de "dissociation","dephosphorylation","activation_dephosphorylation", "state","activation_indirect","inhibit "expression_indirect","indirect_inhibition","repression", "binding/association_phosphorylation","dissoci If set to null, beta will be by default chosen as: c(1,0,0,1,1,1,0,0,1,1,0,0,1,0,1,1,0,1,1,1,0,0,0).

Value

Get one Data in data frame format,which cotains pathway's id,pathway's name and PFS_score.

---

path_cox_data *path_cox_data*

Description

path_cox_data

Usage

```
path_cox_data
```

Format

An object of class `data.frame` with 105 rows and 9 columns.

`path_gene` *path_gene*

Description

path_gene

Usage

```
path_gene
```

Format

An object of class `list` of length 7.

`path_Ucox_mul` *path_Ucox_mul*

Description

path_Ucox_mul

Usage

```
path_Ucox_mul
```

Format

An object of class `matrix` (inherits from `array`) with 7 rows and 5 columns.

`path_Ucox_mul_res` *path_Ucox_mul_res*

Description

path_Ucox_mul_res

Usage

```
path_Ucox_mul_res
```

Format

An object of class `matrix` (inherits from `array`) with 7 rows and 5 columns.

`pfs_score` *pfs_score*

Description

pfs_score.

Usage

```
pfs_score
```

Format

An object of class `matrix` (inherits from `array`) with 123 rows and 105 columns.

`response` *response*

Description

response.

Usage

```
response
```

Format

An object of class `data.frame` with 110 rows and 2 columns.

`roc_data` *roc_data, the data frame use for ploting ROC curve*

Description

The roc_data is used to generate ROC curves.

Usage

```
roc_data
```

Format

An object of class `matrix` (inherits from `array`) with 105 rows and 4 columns.

sig *sig*

Description

sig

Usage

```
sig
```

Format

An object of class character of length 7.

sur *sur*

Description

sur

Usage

```
sur
```

Format

An object of class data.frame with 110 rows and 2 columns.

symbol_Entrez *symbol_Entrez*

Description

symbol_Entrez

Usage

```
symbol_Entrez
```

Format

An object of class data.frame with 54245 rows and 2 columns.

Index