Package 'iATMEcell'

August 24, 2022

Type Package

Title Identification of Abnormal Tumor Microenvironment Cells

Version 0.1.0

Maintainer Junwei Han <hanjunwei1981@163.com>

Description A systematic biology tool was developed to identification of abnormal tumor microenvironment cells. iATMEcell first construct a cell-cell crosstalk network based on cell functions, and then it used a network propagation algorithm to identify significantly abnormal cells and verify their prognostic efficacy.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Depends R (>= 3.6)

RoxygenNote 7.1.1

Imports forestplot,
ggplot2,
ggpubr,
igraph,
pheatmap,
plyr,
reshape2,
stats,
survival,
survminer,
glmnet

Suggests rmarkdown,
knitr

VignetteBuilder knitr

R topics documented:

`envData` *An environment variable that includes some example data*

Description

An environment variable that includes some example data. GEP: An example gene expression profile; TME_related_Goterm: Biological process data from Gene Ontology; GoCellconGene: Gene symbols shared by a pair of cell and biological process (GOterm); Jaccardscore: Jaccard score calculated based on genes shared by a pair of cell and biological process (GOterm); clinicaldata: Clinical information of samples in gene expression profile; Condition.label: Classification information of samples in gene expression profile; TMEcellinfo: TME cells information.

Usage

```
envData
```

Format

An environment variable

Details

The biological function data was derived from GO biological processes, In the "gene ontology" term, a biological process represents a specific objective that the organism is genetically programmed to achieve [23]. The biological process gene sets were downloaded from C5 GO gene sets in the Molecular Signatures Database (MSigDB) database (version 7.0) [24]. We then manually curated the GO gene sets associated with human immune function, and obtained 139 GO terms, which were deposited in our "iATMEcell" package.

`GetExampleSet` *Get example dataset*

Description

This function is used to achieve example dataset.

Usage

```
GetExampleSet(exampleData)
```

Arguments

`exampleData`	A character, should be one of"GEP","clinicaldata", "Condition.label" "TMEcellinfo", "TME_related_Goterm", "Jaccardscore" and "GoCellconGene".

Value

example dataset

iTMEcell *Identification of abnormal tumor microenvironment (TME) cells*

Description

The function "iTMEcell" is used to calculate the eigenvector centrality of TME cells and identify abnormal TME cells.

Usage

```
iTMEcell(ExpData,Condition.label,nperm=1000)
```

Arguments

ExpData	A gene expression profile of interest (rows are genes, columns are samples).
Condition.label	A data.frame at least two columns which are “sample” (sample.id) and ”Condition.label” (condition of samples, “0” represents case and “1” represents control).
nperm	Number of random permutations (default: 1000).

Value

A dataframe with seven columns those are cell names, marker source, marker size, marker genes, centrality (eigenvector centrality), P-value and FDR.

Examples

```
library(igraph)
#Obtain input data
GEP<-GetExampleSet('GEP')
Condition.label<-GetExampleSet('Condition.label')
#Run the function
iTMEcellresult<-iTMEcell(ExpData=GEP,Condition.label=Condition.label,nperm=1000)
```

plotforest *Draw a forest plot.*

Description

The function "plotforest" is used to draw a forest plot according to the result of cox analysis from function "RiskRegressModel".

Usage

```
plotforest(Regress.list,p.cutoff=0.05,g.pos=2,b.size=3,
        col=c("#FE0101","#1C61B6","#A4A4A4"),
        lwd.zero=2,lwd.ci=3,x.lab="Hazard Ratio Plot")
```

Arguments

Regress.list	The result of function "RiskRegressModel".
p.cutoff	Statistical significance threshold of cox regression analysis, based on which to determine the genes used to draw the graph.
g.pos	A number to control the position of the graph element in forestplot.
b.size	A number to control the box size.
col	Vector of colors including three color code which are corresponding to box, box line and reference line.
lwd.zero	A number to control the thickness of the reference line.
lwd.ci	A number to control the thickness of the box line.
x.lab	Setting the title.

Value

A forest plot

Examples

```
library(forestplot)
library(survival)
#Obtain input data
GEP<-GetExampleSet('GEP')
clinicaldata<-GetExampleSet('clinicaldata')
#Run the function
R.result<-RiskRegressModel(cellname='NK cells',ExpData=GEP,clinical=clinicaldata,
     p.cutoff=0.05)
plotforest(Regress.list=R.result,p.cutoff=0.05)
```

---

plotHeatmap *Draw a heat map.*

Description

The function "plotHeatmap" is used to draw a heat map of marker genes.

Usage

```
plotHeatmap(Regress.list,ExpData,cut.off=NULL,p.cutoff=0.05,cluster.rows=F,
cluster.cols=F,bk=c(-2.4,2.3),show.rownames=T,show.colnames=F,
ann_colors=c("#FFAA2C","#2CBADA"),col=c("#2A95FF","#FF1C1C"))
```

Arguments

Regress.list	The result of function "RiskRegressModel".
ExpData	A gene expression profile of interest (rows are genes, columns are samples).
cut.off	Samples will be grouped according to this threshold. If not specified, the sample will be grouped according to the median risk score.
p.cutoff	Statistical significance threshold of cox regression analysis, based on which to determine the genes used to draw the graph.

cluster.rows	Boolean values determining if rows should be clustered or hclust object.
cluster.cols	Boolean values determining if columns should be clustered or hclust object.
bk	A numeric vector that covers the range of values. Users could adjust color depth through this parameter.
show.rownames	Boolean specifying if row names are be shown.
show.colnames	Boolean specifying if column names are be shown.
ann_colors	Vector of colors for specifying the color of column annotation.
col	Vector of colors used in heat map.

Value

A heat map

Examples

```
library(pheatmap)
library(survival)
#Obtain input data
GEP<-GetExampleSet('GEP')
clinicaldata<-GetExampleSet('clinicaldata')
#Run the function
R.result<-RiskRegressModel(cellname='NK cells',ExpData=GEP,clinical=clinicaldata,
      p.cutoff=0.05)
plotHeatmap(Regress.list=R.result,ExpData=GEP,p.cutoff=0.05)
```

---

plotKMcurve *Draw a Kaplan-Meier curve.*

Description

The function "plotKMcurve" is used to draw the Kaplan-Meier curve according to the riskscore of samples from function "RiskRegressModel".

Usage

```
plotKMcurve(Regress.list,ExpData,risk.table=TRUE,labs=c("High risk","Low risk"),
                  title="Group",line.col=c("#FFAA2C", "#2CBADA"))
```

Arguments

Regress.list	The result of function "RiskRegressModel".
ExpData	A gene expression profile of interest (rows are genes, columns are samples).
risk.table	TRUE or FALSE specifying whether to show or not the risk table. Default is TRUE.
labs	A character vector for specifying legend labels.
title	legend title
line.col	Vector of colors for specifying the color of curve.

Value

Kaplan-Meier curve

Examples

```
library(survival)
library(survminer)
#Obtain input data
GEP<-GetExampleSet('GEP')
clinicaldata<-GetExampleSet('clinicaldata')
#Run the function
R.result<-RiskRegressModel(cellname='NK cells',ExpData=GEP,clinical=clinicaldata,
       p.cutoff=0.05)
plotKMcurve(Regress.list=R.result,ExpData=GEP)
```

`plotSplitViolin` *Draw a split violin plot.*

Description

The function "plotSplitViolin" is used to draw a split violin plot of gene expression.

Usage

```
plotSplitViolin(Regress.list,ExpData,gene.name,method="t.test",
compare.label="p.signif",col=c("#E69F00", "#56B4E9"),x.ceiling=15,
y.lab="Gene Expression",x.lab=NULL,title=NULL)
```

Arguments

`Regress.list`	The result of function "RiskRegressModel".
`ExpData`	A gene expression profile of interest (rows are genes, columns are samples).
`gene.name`	A gene symbol in inputted gene expression profile.
`method`	A character string indicating which method to be used for comparing means. The default method is "t.test". Other three methods are "wilcox.test", "anova" and "kruskal.test".
`compare.label`	A character string specifying label type. Allowed values include "p.signif" (shows the significance levels), "p.format" (shows the formatted p value).
`col`	Vector of colors used to specify the color of different groups.
`x.ceiling`	The maximum value of the y axis.
`y.lab`	Setting the title of the y-axis.
`x.lab`	Setting the title of the x-axis.
`title`	Setting the title

Value

A split violin plot

Examples

```
library(ggplot2)
library(reshape2)
library(plyr)
library(ggpubr)
library(survival)
#Obtain input data
GEP<-GetExampleSet('GEP')
clinicaldata<-GetExampleSet('clinicaldata')
#Run the function
R.result<-RiskRegressModel(cellname='NK cells',ExpData=GEP,clinical=clinicaldata,
        p.cutoff=0.05)
plotSplitViolin(Regress.list=R.result,ExpData=GEP,gene.name="PDCD1")
```

`RiskRegressModel` *Constructing the cox risk regression model with cell's marker genes*

Description

This function is used to perform regression analysis and build risk regression models.

Usage

```
RiskRegressModel(cellname,ExpData,clinical,marker=NULL,method = "lasso",p.cutoff=0.05)
```

Arguments

`cellname`	A cell whose marker genes will be used to perform regression analysis. The format of the entered cell name should refer to the cell information we provide.
`ExpData`	A gene expression profile of interest (rows are genes, columns are samples).
`clinical`	A dataframe with three columns which are "sample" (sample id),"status" (survival status of samples, "0" represents live and "1" represents dead) and "time" (survival time of samples).
`marker`	A character vector composed with marker genes. If you does not want to use the marker genes provided by us, you can specify the marker genes you need with this parameter.
`method`	This parameter specifies the method of regression analysis. "method=cox": Only univariate regression analysis was performed, and the model was constructed with coefficients. "method=lasso"(default): The significant variables in univariate analysis were used for LASSO regression analysis, and the coefficients of LASSO analysis were used to construct the model.
`p.cutoff`	Statistical significance threshold for regression analysis (default: 0.05).

Details

In the default method, users can specify a cell, and then the function will perform cox regression analysis on expression value of marker genes of the cell and survival data. Statistical significant genes will be selected for further lasso regression analysis. Finally, the lasso regression coefficients were used to weight the gene expression values to calculate the risk score for samples.

Value

A list with two dataframes which are riskscores of samples and result of cox regression analysis respectively.

Examples

```
library(survival)
#Obtain input data
GEP<-GetExampleSet('GEP')
clinicaldata<-GetExampleSet('clinicaldata')
#Run the function
R.result<-RiskRegressModel(cellname='NK cells',ExpData=GEP,
  clinical=clinicaldata,method = "lasso",p.cutoff=0.05)
```

Index