Package 'SMDIC'

January 20, 2025

Type Package

Title Identification of Somatic Mutation-Driven Immune Cells

Version 0.1.6

Maintainer Junwei Han <hanjunwei1981@163.com>

Description A computing tool is developed to automated identify somatic mutation-driven immune cells. The operation modes including: i) inferring the relative abundance matrix of tumor-infiltrating immune cells and integrating it with a particular gene mutation status, ii) detecting differential immune cells with respect to the gene mutation status and converting the abundance matrix of significant differential immune cell into two binary matrices (one for up-regulated and one for down-regulated), iii) identifying somatic mutation-driven immune cells by comparing the gene mutation status with each immune cell in the binary matrices across all samples, and iv) visualization of immune cell abundance of samples in different mutation status..

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Depends R (>= 4.3.0), GSVA

Imports samr, e1071, parallel, preprocessCore, pheatmap, maftools, grDevices, survival, survminer, MASS, pracma, stats, RColorBrewer, backports

Suggests knitr, rmarkdown, R.utils

VignetteBuilder knitr

NeedsCompilation no

Author Junwei Han [cre],
Baotong Zheng [aut]

Repository CRAN

Date/Publication 2024-10-16 09:10:01 UTC

Contents

`SMDIC-package` *Identification of somatic mutation-driven immune cells*

Description

With the use of functions in this packages, users could identify the immune cells driven by somatic mutations in tumor microenvironment.

`cell24` *A data.frame of 24 immune cells name from Bindea et al*

Description

It's a built-in data. The first column represents the abbreviation of 24 immune cells, the second column represents the full name of 24 immune cells

Usage

```
cell24
```

Format

A data.frame with 24 rows and 2 column

References

Bindea G, Mlecnik B, Tosolini M, Kirilovsky A, Waldner M, Obenauf AC, et al. Spatiotemporal dynamics of intratumoral immune cells reveal the immune landscape in human cancer. Immunity. 2013;39:782–95.

`cell64` *A data.frame of 64 immune cells name from xCell method*

Description

It's a built-in data. The first column represents the abbreviation of 64 immune cells, the second column represents the full name of 64 immune cells

Usage

```
cell64
```

Format

A data.frame with 64 rows and 2 column

References

Aran D , Hu Z , Butte A J . xCell: digitally portraying the tissue cellular heterogeneity landscape[J]. Genome Biology, 2017, 18(1):220.

`envData` *envData*

Description

The variables in the environment include an example expression profiles, a cell abundance matrix, a binary numerical matrix which shows the immune cells driven by somatic mutation, a binary mutations matrix.

Format

An environment variable

Details

The environment variable includes the variable `exp.example`, `cellmatrix`,`mutcell`,`mutmatrix`

Author(s)

Junwei Han<`hanjunwei1981@163.com`>,Baotong Zheng<`btzheng1116@163.com`>

exp2cell *exp2cell*

Description

Function ‘exp2cell‘ use gene expression profiles to quantify cell abundance matrix. ‘exp2cell‘ provides three methods for estimating the relative infiltration abundance of different cell types in the tumor microenvironment (TME), which including xCell, ssGSEA estimated method proposed by Şenbabaoğlu et al. and CIBERSORT.

Usage

```
exp2cell(exp, method = "xCell", perm = 100, QN = TRUE)
```

Arguments

`exp`	The gene expression data set. A matrix with row names as symbols and columns as samples. Gene expression profiles were used to quantify cell abundance matrix.
`method`	Method must be one of "xCell", "ssGSEA" and "CIBERSORT".
`perm`	No. permutations; set to >=100 to calculate p-values (default = 100)
`QN`	Quantile normalization of input mixture (default = TRUE)

Value

Cell abundance matrix.

References

1. Aaron, M, Newman, et al. Robust enumeration of cell subsets from tissue expression profiles.[J]. Nature Methods, 2015. 2. Aran D , Hu Z , Butte A J . xCell: digitally portraying the tissue cellular heterogeneity landscape[J]. Genome Biology, 2017, 18(1):220. 3. Şenbabaoğlu, Yasin, Gejman R S , Winer A G , et al. Tumor immune microenvironment characterization in clear cell renal cell carcinoma identifies prognostic and immunotherapeutically relevant messenger RNA signatures[J]. Genome biology, 2016, 17(1).

Examples

```
#get breast cancer gene expression profile.
exp.example<-GetExampleData("exp.example")

#perform the exp2cell method. Method must be one of "xCell","ssGSEA" and "CIBERSORT".
cellmatrix<-exp2cell(exp=exp.example,method="ssGSEA") #cell abundance matrix
```

`gene2cellsummary` *gene2cellsummary*

Description

Function ‘gene2cellsummary‘ is a generic function used to produce result summaries of the immune cells driven by a somatic mutation.

Usage

```
gene2cellsummary(gene, method = ”xCell”, mutcell)
```

Arguments

`gene`	Somatic mutant gene name
`method`	Method must be one of "xCell","ssGSEA" and "CIBERSORT".
`mutcell`	The result of ‘mutcorcell‘ funtion.

Value

A matrix shows the short name, full name, pvalue, fdr, cell responses(up or down) of the cells driven by a somatic mutation.

Examples

```
# get the result of `mutcorcell` funtion.
mutcell<-GetExampleData(”mutcell”)

# perform the function gene2cellsummary
genecellsummary<-gene2cellsummary(gene=”TP53”,mutcell=mutcell)
```

`GetExampleData` *Get the example data*

Description

Get the example data from SMDIC package.

Usage

```
GetExampleData(exampleData)
```

Arguments

`exampleData`	A character, should be one of "exp.example", "cellmatrix", "mutcell", "mutmatrix", "surv".

Details

The function ‘GetExampleData(ExampleData = "mutmatrix)")‘ obtains the mutations matrix

References

Subramanian, A., Tamayo, P., Mootha, V.K., Mukherjee, S., Ebert, B.L., Gillette, M.A., Paulovich, A., Pomeroy, S.L., Golub, T.R., Lander, E.S. et al. (2005) Gene set enrichment analysis: a knowledgebased approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci U S A, 102, 15545-15550.

---

heatmapcell *heatmapcell*

Description

A function to draw clustered heatmaps for the cells driven by a somatic mutation.

Usage

```
heatmapcell(
  gene,
  mutcell,
  cellmatrix,
  mutmatrix,
  title = NA,
  show_rownames = TRUE,
  show_colnames = FALSE,
  annotation_colors = NA,
  annotation_row = NA,
  annotation_col = NA,
  color = colorRampPalette(rev(brewer.pal(n = 7, name = "RdYlBu")))(100)
)
```

Arguments

`gene`	Somatic mutant gene name
`mutcell`	A list, mutcell is the result of function ‘mutcorcell‘.
`cellmatrix`	Cell abundance matrix, cellmatrix is the result of function ‘exp2cell‘.
`mutmatrix`	A binary mutations matrix, which can not only come from the maf2matrix function, but also any binary mutations matrix, in which 1 represents any mutation occurs in a particular gene in a particular sample, otherwise the element is 0.
`title`	The title of the plot
`show_rownames`	boolean specifying if column names are be shown.
`show_colnames`	boolean specifying if column names are be shown.

`annotation_colors`
list for specifying annotation_row and annotation_col track colors manually. It is possible to define the colors for only some of the features. Check examples for details.

`annotation_row` data frame that specifies the annotations shown on left side of the heatmap. Each row defines the features for a specific row. The rows in the data and in the annotation are matched using corresponding row names. Note that color schemes takes into account if variable is continuous or discrete.

`annotation_col` similar to annotation_row, but for columns.

`color` vector of colors used in heatmap.

Examples

```
#get the result of `mutcorcell` function.
mutcell<-GetExampleData("mutcell")

#get cell abundance matrix which is the result of exp2cell function
cellmatrix<-GetExampleData("cellmatrix")

#get the binary mutations matrix
mutmatrix<-GetExampleData("mutmatrix")

# plot significant up-regulation or down-regulation cells heat map specific for breast cancer
heatmapcell(gene = "TP53",mutcell = mutcell,cellmatrix = cellmatrix,mutmatrix = mutmatrix)
```

`immunelist` *A large list of 24 immune cells type-specific gene signatures from Bindea et al*

Description

It's a built-in data. The name of the list represent 24 immune cells, the value of the list are 24 immune cells type-specific gene signatures from Bindea et al

Usage

```
immunelist
```

Format

A list

References

Bindea G, Mlecnik B, Tosolini M, Kirilovsky A, Waldner M, Obenauf AC, et al. Spatiotemporal dynamics of intratumoral immune cells reveal the immune landscape in human cancer. Immunity. 2013;39:782–95.

maf2matrix *maf2matrix*

Description

Function ‘maf2matrix‘ use mutation annotation file (MAF) format data to build a binary mutations matrix.

Usage

```
maf2matrix(maffile, percent = 0.01, nonsynonymous = TRUE)
```

Arguments

`maffile` The name of mutation annotation file (MAF) format data. It must be an absolute path or the name relatived to the current working directory.

`percent` A threshold value(one percent as the default value). The genes with a given mutation frequency equal or greater than the threshold value are retained for the following analysis.

`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 any mutation occurs in a particular gene in a particular sample, otherwise the element is 0.

Examples

```
#get path of the mutation annotation file.
maf = system.file('extdata', 'example.maf.gz', package = 'SMDIC')

# perform function `maf2matrix`.
mutmatrix.example<-maf2matrix(maf)
```

mutcellsummary *mutcellsummary*

Description

Function ‘mutcellsummary‘ is a generic function used to produce summaries of the results of ‘mutcorcell‘ function.

Usage

```
mutcellsummary(mutcell, mutmatrix, cellmatrix)
```

Arguments

`mutcell`	The result of ‘mutcorcell‘ funtion.
`mutmatrix`	A binary mutations matrix, which can not only come from the maf2matrix function, but also any binary mutations matrix, in which 1 represents any mutation occurs in a particular gene in a particular sample, otherwise the element is 0.
`cellmatrix`	Cell abundance matrix

Value

The result summaries have four columns. The first column is somatic mutant gene names, the second column is the immune cell names driven by the somatic mutation, the third column is the number of the immune cell, the fourth column is the mutation rate.

Examples

```
# get result of `mutcorcell` funtion
mutcell<-GetExampleData("mutcell")

#get cell abundance matrix which is the result of exp2cell function
cellmatrix<-GetExampleData("cellmatrix")

# get the binary mutations matrix
mutmatrix<-GetExampleData("mutmatrix") # A binary mutations matrix

#perform the function mutcellsummary
summary<-mutcellsummary(mutcell = mutcell,mutmatrix = mutmatrix,cellmatrix=cellmatrix)
```

`mutcorcell` *mutcorcell*

Description

Function ‘mutcorcell‘ identifies somatic mutation-driven immune cells by comparing the cell abundance matrix and binary mutations matrix.

Usage

```
mutcorcell(
  cellmatrix = cellmatrix,
  mutmatrix = mutmatrix,
  samfdr.cutoff = 0.05,
  nperms = 100,
  fisher.cutoff = 0.05,
  fisher.adjust = FALSE
)
```

Arguments

`cellmatrix` Cell abundance matrix.

`mutmatrix` A binary mutations matrix, which can not only come from the maf2matrix function, but also any binary mutations matrix, in which 1 represents any mutation occurs in a particular gene in a particular sample, otherwise the element is 0.

`samfdr.cutoff` False Discovery Rate cutoff for output in significant immune cells

`nperms` Number of permutations used by SAM to estimate False Discovery Rates

`fisher.cutoff` False Discovery Rate(fisher.adjust=TRUE) or P-Value(fisher.adjust=FALSE) cutoff for Fisher's exact test

`fisher.adjust` Logical,tell if corrects p-values

Value

A list of four matrices: a binary numerical matrix which shows the immune cells driven by somatic mutant gene; two numerical matrix which show the pvalue and fdr of the immune cells driven by somatic mutant gene; a character matrix which shows the cell responses of the immune cells driven by somatic mutant gene.

Examples

```
#get cell abundance matrix which is the result of exp2cell function
cellmatrix<-GetExampleData("cellmatrix")

#get the binary mutations matrix,
mutmatrix<-GetExampleData("mutmatrix")

#perform the function `mutcorcell`.
mutcell<-mutcorcell(cellmatrix = cellmatrix,mutmatrix = mutmatrix)

# The summary for somatic mutations are produced by function `mutcellsummary`.
#summary<-mutcellsummary(mutcell = mutcell,mutmatrix = mutmatrix,cellmatrix=cellmatrix)

# The summary of the immune cells driven by a mutation are produced by function `gene2cellsummary`.
#genecellsummary<-gene2cellsummary(gene="TP53",mutcell=mutcell)
```

`plotCoocMutex` *plotCoocMutex*

Description

Function 'plotCoocMutex' plots the co-occurrence and mutual exclusivity plots for mutation genes which drive immune cells.

Usage

```
plotCoocMutex(maffile, mutcell.summary, cellnumcuoff = 3, fontSize = 0.8)
```

Arguments

`maffile` The name of mutation annotation file (MAF) format data. It must be an absolute path or the name relatived to the current working directory.

`mutcell.summary` The result of ‘mutcellsummary‘ function

`cellnumcuoff` A threshold value (4 as the default value). The mutation genes which drive at least "cellnumcuoff" cells are retained for drawing a co-occurrence and mutual exclusivity plots.

`fontSize` cex for gene names. Default 0.8

References

Gerstung M, Pellagatti A, Malcovati L, et al. Combining gene mutation with gene expression data improves outcome prediction in myelodysplastic syndromes. Nature Communications. 2015;6:5901. doi:10.1038/ncomms6901.

Examples

```
# get the result of `exp2cell` funtion
cellmatrix<-GetExampleData("cellmatrix")

#get the binary mutations matrix,
mutmatrix<-GetExampleData("mutmatrix")

# get the result of `mutcorcell` funtion
mutcell<-GetExampleData("mutcell")

#perform the function mutcellsummary
summary<-mutcellsummary(mutcell = mutcell,mutmatrix = mutmatrix,cellmatrix=cellmatrix)

#dir is the name of mutation annotation file (MAF) format data.
#It must be an absolute path or the name relatived to the current working directory.
maf<-system.file("extdata", "example.maf.gz", package = "SMDIC") #MAF file
#plot the co-occurrence and mutual exclusivity plots for mutation genes which drive immune cells.
plotCoocMutex(maffile = maf,mutcell.summary = summary,cellnumcuoff =0)
```

`plotwaterfall` *plotwaterfall*

Description

Function ‘plotwaterfall‘ plots the waterfall for mutation genes which drive immune cells.

Usage

```
plotwaterfall(
  maffile,
  mutcell.summary,
  cellnumcuoff = 3,
  fontSize = 0.8,
  showTumorSampleBarcodes = F,
  showTitle = TRUE,
  colors = NULL
)
```

Arguments

`maffile`	The name of mutation annotation file (MAF) format data. It must be an absolute path or the name relatived to the current working directory.
`mutcell.summary`	The result of ‘mutcellsummary‘ function
`cellnumcuoff`	a threshold value (3 as the default value). The mutation genes which drive at least "cellnumcuoff" cells are retained for drawing an waterfall.
`fontSize`	font size for gene names. Default 0.8.
`showTumorSampleBarcodes`	logical to include sample names.
`showTitle`	Default TRUE
`colors`	named vector of colors for each Variant_Classification.

Examples

```
# get result of `exp2cell` funtion
cellmatrix<-GetExampleData("cellmatrix")

#get the binary mutations matrix,
mutmatrix<-GetExampleData("mutmatrix")

# get the result of `mutcorcell` funtion
mutcell<-GetExampleData("mutcell")

#perform the function mutcellsummary
summary<-mutcellsummary(mutcell = mutcell,mutmatrix = mutmatrix,cellmatrix=cellmatrix)

#dir is the name of mutation annotation file (MAF) format data.
#It must be an absolute path or the name relatived to the current working directory.
maf<-system.file("extdata", "example.maf.gz", package = "SMDIC") #MAF file
# mutcell.summary is the result of function mutcellsummary

#plot the waterfall for mutation genes which drive immune cells
plotwaterfall(maffile = maf,mutcell.summary = summary,cellnumcuoff =3)
```

survcell *survcell*

Description

Function ‘survcell‘ draws Kaplan–Meier curves for survival in the above-median and below-median groups for cell risk score. The cell risk score is calaulated by the weighted mean of cells driven by a gene mutation, where the weight of cells is estimated by the "Univariate" or "Multivariate" cox.

Usage

```
survcell(
  gene,
  mutcell,
  cellmatrix,
  surv,
  method = ”Multivariate”,
  legend.title = ”Strata”,
  legend.labs = c(”group=0”, ”group=1”),
  palette = c(”#E7B800”, ”#2E9FDF”),
  color = NULL,
  pval = TRUE,
  title = NULL,
  ggtheme = theme_survminer()
)
```

Arguments

`gene` Somatic mutant gene name

`mutcell` The result of ‘mutcorcell‘ function

`cellmatrix` Cell abundance matrix

`surv` Surv is the survival data, the first column is the sample name, the second column is the survival time, and the third is the survival event.

`method` Method must be one of "Univariate" and "Multivariate". The coefficient of cells for risk score are estimated by "Univariate" or "Multivariate" cox proportional risk regression model on cell abundance matrix and overall survival data..

`legend.title` legend title.

`legend.labs` character vector specifying legend labels. Used to replace the names of the strata from the fit. Should be given in the same order as those strata.

`palette` the color palette to be used. Allowed values include "hue" for the default hue color scale; "grey" for grey color palettes; brewer palettes e.g. "RdBu", "Blues", ...; or custom color palette e.g. c("blue", "red"); and scientific journal palettes from ggsci R package, e.g.: "npg", "aaas", "lancet", "jco", "ucscgb", "uchicago", "simpsons" and "rickandmorty". See details section for more information. Can be also a numeric vector of length(groups); in this case a basic color palette is created using the function palette.

`color` color to be used for the survival curves.If the number of strata/group (n.strata) = 1, the expected value is the color name. For example color = "blue".If n.strata > 1, the expected value is the grouping variable name. By default, survival curves are colored by strata using the argument color = "strata", but you can also color survival curves by any other grouping variables used to fit the survival curves. In this case, it's possible to specify a custom color palette by using the argument palette.

`pval` logical value, a numeric or a string. If logical and TRUE, the p-value is added on the plot. If numeric, than the computet p-value is substituted with the one passed with this parameter. If character, then the customized string appears on the plot.

`title` the title of the survival curve

`ggtheme` function, ggplot2 theme name. Default value is theme_survminer. Allowed values include ggplot2 official themes: see theme.

Value

Kaplan–Meier curves

Examples

```
# get the result of `mutcorcell` function.
mutcell<-GetExampleData("mutcell")

# get cell abundance matrix which is the result of exp2cell function
cellmatrix<-GetExampleData("cellmatrix")

# get survival data
surv<-GetExampleData("surv")

#draw Kaplan-Meier curves
survcell(gene ="TP53",mutcell=mutcell,cellmatrix=cellmatrix,surv=surv)
```

`xCell.data` *xCell datasets*

Description

xCell datasets. It's a built-in data.

Usage

```
xCell.data
```

Format

list:

spill spillover matrix and calibration parameters

signatures the signatures for calculating scores

genes genes to use to calculate xCell

Index