Package ‘iPRISM’

July 22, 2025

Type Package

Title Intelligent Predicting Response to Cancer Immunotherapy Through Systematic Modeling

Version 0.1.1

Maintainer Junwei Han <`hanjunwei1981@163.com`>

Description Immunotherapy has revolutionized cancer treatment, but predicting patient response remains challenging. Here, we presented Intelligent Predicting Response to cancer Immunotherapy through Systematic Modeling (iPRISM), a novel network-based model that integrates multiple data types to predict immunotherapy outcomes. It incorporates gene expression, biological functional network, tumor microenvironment characteristics, immune-related pathways, and clinical data to provide a comprehensive view of factors influencing immunotherapy efficacy. By identifying key genetic and immunological factors, it provides an insight for more personalized treatment strategies and combination therapies to overcome resistance mechanisms.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

Imports ggplot2, Hmisc, tidyr, igraph, pbapply, Matrix, methods

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Junwei Han [aut, cre, ctb],
Yinchun Su [aut],
Siyuan Li [aut]

Repository CRAN

Date/Publication 2024-07-14 12:00:02 UTC

Contents

`iPRISM-package` *Intelligent Predicting Response to Cancer Immunotherapy Through Systematic Modeling*

Description

Immunotherapy has revolutionized cancer treatment, but predicting patient response remains challenging. Here, we presented Intelligent Predicting Response to cancer Immunotherapy through Systematic Modeling (iPRISM), a novel network-based model that integrates multiple data types to predict immunotherapy outcomes. It incorporates gene expression, biological functional network, tumor microenvironment characteristics, immune-related pathways, and clinical data to provide a comprehensive view of factors influencing immunotherapy efficacy. By identifying key genetic and immunological factors, it provides an insight for more personalized treatment strategies and combination therapies to overcome resistance mechanisms.

Details

iPRISM

Examples

```
# if (!"devtools" %in% as.data.frame(installed.packages())$Package)
#   install.packages("devtools")
# devtools::install_github("hanjunwei-lab/iPRISM")
```

cor_plot *Correlation Plot with Significance Points*

Description

This function generates a correlation plot between two datasets, displaying correlation coefficients as a heatmap and significant correlations as scatter points.

Usage

```
cor_plot(
  data1,
  data2,
  sig.name1 = "value1",
  sig.name2 = "value2",
  cutoff.pvalue = 0.05,
  color = c("#62CCC9", "#FF9999")
)
```

Arguments

`data1`	A data frame or matrix representing the first dataset.
`data2`	A data frame or matrix representing the second dataset.
`sig.name1`	A character string specifying the name of the first dataset (default: "value1").
`sig.name2`	A character string specifying the name of the second dataset (default: "value2").
`cutoff.pvalue`	The significance threshold for correlation (default: 0.05).
`color`	A vector of two colors for the heatmap gradient (default: c("#62CCC9", "#FF9999")).

Details

The function computes correlation coefficients between corresponding columns in the two datasets and identifies significant correlations based on p-values.

Value

A ggplot object displaying the correlation heatmap and scatter points.

Examples

```
# Read all data into memory
data(data.path, package = "iPRISM")
data(data.cell, package = "iPRISM")
# Draw the plot
cor_plot(data1 = data.path,data2 = data.cell,sig.name1 = "path",sig.name2 = "cell")
```

`data.cell` *data.cell*

Description

The ‘data.cell‘ represents the second type of feature matrix used for calculating correlations (in this case, cell abundances), where rows correspond to samples, and columns correspond to features.

Usage

```
data.cell
```

Format

An object of class `matrix` (inherits from `array`) with 121 rows and 21 columns.

Examples

```
library(iPRISM)
data(data.cell, package = "iPRISM")
dim(data.cell)
```

`data.path` *data.path*

Description

The ‘data.path‘ represents the first type of feature matrix used for calculating correlations (in this case, pathway expression levels), where rows correspond to samples, and columns correspond to features.

Usage

```
data.path
```

Format

An object of class `matrix` (inherits from `array`) with 121 rows and 17 columns.

Examples

```
library(iPRISM)
data(data.path, package = "iPRISM")
plot(data.path)
```

`data_sig` *data_sig*

Description

The 'data_sig' represents the sample feature matrix, where rows correspond to samples, and columns correspond to features.

Usage

```
data_sig
```

Format

An object of class `matrix` (inherits from `array`) with 121 rows and 31 columns.

Examples

```
library(iPRISM)
data(data_sig, package = "iPRISM")
plot(data_sig)
```

`ESscore` *Enrichment Score Calculation*

Description

Calculates the enrichment score (ES) for a given set of labels and correlation vector.

Usage

```
ESscore(labels.list, correl.vector = NULL)
```

Arguments

`labels.list`	A binary vector indicating membership in a gene set (1 for inclusion, 0 for exclusion).
`correl.vector`	A vector of correlation values (e.g., gene expression correlations).

Value

The enrichment score (ES) for the given labels and correlation vector.

`ESscore_weighted` *Weighted Enrichment Score Calculation*

Description

Calculates the weighted enrichment score (ES) for a given set of labels and correlation vector.

Usage

```
ESscore_weighted(labels.list, correl.vector = NULL)
```

Arguments

`labels.list`	A binary vector indicating membership in a gene set (1 for inclusion, 0 for exclusion).
`correl.vector`	A vector of correlation values (e.g., gene expression correlations).

Value

The weighted enrichment score (ES) for the given labels and correlation vector.

`genelist_cp` *TME gene list after random walks*

Description

This gene list includes genes from tumor microenvironment (TME). Random Walk with Restart (RWR) is applied to prioritize genes that are relevant to immunotherapy responses.

Usage

```
genelist_cp
```

Format

An object of class `numeric` of length 15867.

Examples

```
library(iPRISM)
data(genelist_cp, package = "iPRISM")
```

`genelist_hla` *HLA gene list after random walks*

Description

This gene list includes genes from human leukocyte antigen (HLA). Random Walk with Restart (RWR) is applied to prioritize genes that are relevant to immunotherapy responses.

Usage

```
genelist_hla
```

Format

An object of class `numeric` of length 15867.

Examples

```
library(iPRISM)
data(genelist_hla, package = "iPRISM")
```

`genelist_imm` *ICI gene list after random walks*

Description

This gene list includes genes from immune checkpoint inhibitors (ICI). Random Walk with Restart (RWR) is applied to prioritize genes that are relevant to immunotherapy responses.

Usage

```
genelist_imm
```

Format

An object of class `numeric` of length 15867.

Examples

```
library(iPRISM)
data(genelist_imm, package = "iPRISM")
```

`get_gsea_path` *Gene Set Enrichment Analysis (GSEA) using Multiplex Networks*

Description

This function performs gene set enrichment analysis (GSEA) based on multiplex network data.

Usage

```
get_gsea_path(
  seed = seed,
  network = network,
  gamma = 0.7,
  pathlist = pathlist,
  gsea.weight = 1,
  gsea.nperm = 1000
)
```

Arguments

`seed`	A seed value (optional).
`network`	A network object (e.g., protein-protein interaction network).
`gamma`	A parameter for random walk restart (default: 0.7).
`pathlist`	A predefined list of gene sets (pathways).
`gsea.weight`	Weight for GSEA (default: 1).
`gsea.nperm`	Number of permutations for significance testing (default: 1000).

Details

The function constructs a multiplex network, performs random walk restart, and calculates gene scores. It then transforms the scores and applies GSEA using the provided gene sets.

Value

A GSEA result object.

Examples

```
data(Seeds, package = "iPRISM")
data(ppi, package = "iPRISM")
data(path_list, package = "iPRISM")


result <- get_gsea_path(seed = Seeds,
                        network = ppi,
                        pathlist = path_list[1:2],
                        gsea.nperm = 100)
```

```
print(result)
```

`get_logiModel` *Fit Logistic Regression Model*

Description

This function fits a logistic regression model to the given data.

Usage

```
get_logiModel(data.sig, pred.value, levels = c("R", "N"), step = TRUE)
```

Arguments

`data.sig`	A data frame where each row is a sample and each column is a pathway.
`pred.value`	A numeric vector representing the response variable.
`levels`	A character vector specifying the levels of the response variable (default: c("R", "N")).
`step`	Logical. If TRUE, perform stepwise model selection (default: TRUE).

Details

The function converts the response variable to a factor with specified levels and fits a logistic regression model using the glm function.

Value

A fitted logistic regression model.

Examples

```
data(data_sig, package = "iPRISM")

b <- get_logiModel(data.sig = data_sig, pred.value = pred_value, step = TRUE)
summary(b)
```

`gseafun` *Gene Set Enrichment Analysis (GSEA) Function*

Description

This function performs gene set enrichment analysis using a gene list and a set of pathways.

Usage

```
gseafun(genelist, pathlist, nperm = 1000, weighted = 1)
```

Arguments

`genelist`	A named vector of gene expression values.
`pathlist`	A list of gene sets (pathways) to test for enrichment.
`nperm`	Number of permutations for calculating p-values (default is 1000).
`weighted`	Logical indicating whether to use weighted enrichment scores (default is TRUE).

Value

A data frame with enrichment scores (ES), p-values, and adjusted p-values.

Examples

```
data(path_list, package = "iPRISM")
data(genelist_imm, package = "iPRISM")

res_gsea_imm <- gseafun(genelist = genelist_imm,
                        pathlist = path_list[1:2],
                        weighted = 1,
                        nperm = 1000)
print(res_gsea_imm)
```

`path_list` *path_list*

Description

The ‘path_list‘ contains the gene list associated with pathways.

Usage

```
path_list
```

Format

An object of class `list` of length 2656.

Examples

```
library(iPRISM)
data(path_list, package = "iPRISM")
length(path_list)
```

`ppi` *A protein-protein physical interaction network (PPI network)*

Description

An igraph object containing a protein-protein physical interaction network.

Usage

```
ppi
```

Format

An object of class `igraph` of length 15867.

Examples

```
library(iPRISM)
data(ppi, package = "iPRISM")

library(igraph)
graph <- simplify(ppi)
graph_comp <- components(graph)$membership == which.max(components(graph)$csize)
graph <- induced_subgraph(graph, V(graph)[graph_comp])
plot(graph)
```

`pred_value` *Original Class Labels for Samples*

Description

A named vector where each element corresponds to a sample name and represents the original class label.

A named vector where each element corresponds to a sample name and represents the original class label.

Usage

```
pred_value

pred_value
```

Format

An object of class `character` of length 121.

An object of class `character` of length 121.

Examples

```
library(iPRISM)
data(pred_value, package = "iPRISM")
table(pred_value)

library(iPRISM)
data(pred_value, package = "iPRISM")
table(pred_value)
```

`Seeds` *Seed Node Names*

Description

A character vector with seed node names.

Usage

```
Seeds
```

Format

An object of class `character` of length 3.

Examples

```
library(iPRISM)
data(Seeds, package = "iPRISM")
```

Index