Package ‘EnrichIntersect’

February 26, 2023

Type Package

Title Enrichment Analysis and Intersecting Sankey Diagram

Version 0.6

Date 2023-02-24

URL https://github.com/ocbe-uio/EnrichIntersect

BugReports https://github.com/ocbe-uio/EnrichIntersect/issues

Description

A flexible tool for enrichment analysis based on user-defined sets. It allows users to perform over-representation analysis of the custom sets among any specified ranked feature list, hence making enrichment analysis applicable to various types of data from different scientific fields. 'EnrichIntersect' also enables an interactive means to visualize identified associations based on, for example, the mix-lasso model (Zhao et al., 2022 <doi:10.1016/j.isci.2022.104767>) or similar methods.

License MIT + file LICENSE

Depends R (>= 4.0)

Encoding UTF-8

RoxygenNote 7.2.3

Imports ggplot2, dplyr, networkD3, jsonlite, htmlwidgets, webshot2, stats, methods, grDevices

Suggests knitr

LazyData true

VignetteBuilder knitr

NeedsCompilation no

Author Zhi Zhao [aut, cre],

Manuela Zucknick [aut],

Tero Aittokallio [ctb]

Maintainer Zhi Zhao <zhi.zhao@medisin.uio.no>

Repository CRAN

Date/Publication 2023-02-26 10:40:02 UTC
Data set `cancers_drug_groups` is a list including a score dataframe with 147 drugs as rows and 19 cancer types as columns, and a dataframe with 9 self-defined drug groups (1st column) of the 147 drugs (2nd column). More details of the data can be found in the results of paper 'Tissue-specific identification of multi-omics features for pan-cancer drug response prediction' by Zhao Z., Wang S., Zucknick M. and Aittokallio T. (2022).

Usage

cancers_drug_groups

Format

An object of class list of length 2.

Value

No return value.

References


Examples

# Load data
data(cancers_drug_groups, package = "EnrichIntersect")
Data set 'cancers_genes_drugs' is an array with association scores between 56 genes (1st dimension), three cancer types (2nd dimension) and two drugs (3rd dimension). More details of the data can be found in the results of paper 'Tissue-specific identification of multi-omics features for pan-cancer drug response prediction' by Zhao Z., Wang S. Zucknick M. and Aittokallio T. (2022).

Usage
cancers_genes_drugs

Format
An object of class array of dimension 56 x 3 x 2.

Value
No return value.

References

Examples
# Load data
data(cancers_genes_drugs, package = "EnrichIntersect")

Plot enrichment map through a vector (matrix) of scores and a self-defined set that summarizes a few groups of the names (rownames) of the vector (matrix).
Usage

enrichment(
  x,
  custom.set,
  alpha = 0,
  normalize = TRUE,
  permute.n = 100,
  padj.method = "none",
  pvalue.cutoff = 0.05,
  angle = 45,
  ...
)

Arguments

  x                a vector (matrix) of scores to be enriched
  custom.set      a self-defined set that summarizes a few groups of the names (rownames) of x
  alpha           exponent weight of the score of ordered features. Default is 0 for calculating enrichment score via classic Kolmogorov-Smirnov statistic
  normalize       logic value to determine if normalizing enrichment scores, accounting for custom set size. Default is TRUE
  permute.n       number of custom-set permutations for significance testing. Default is 100
  padj.method     correction method, one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none". Default is "none"
  pvalue.cutoff   a cutoff for both unadjusted and adjusted p-value to mark significantly enriched classes. Default is 0.05
  angle           angle of rotating x-axis labels. Default is 45
  ...             other arguments

Value

Return a list including a matrix of (normalized) enrichment score, a matrix of corresponding p-value and ggplot object:

  • S - a matrix of calculated enrichment scores.
  • pvalue - a matrix of p-values using permutation test for the calculated enrichment scores.
  • g - a ggplot object for visualising the results of an enrichment analysis.

References

**Examples**

# Data set 'cancers_drug_groups' is a list including a score dataframe with 147 drugs as rows and 19 cancer types as columns, and a dataframe with 9 self-defined drug groups (1st column) of the 147 drugs (2nd column).
data(cancers_drug_groups, package = "EnrichIntersect")

x <- cancers_drug_groups$score
custom.set <- cancers_drug_groups$custom.set
set.seed(123)
enrich <- enrichment(x, custom.set, permute.n = 5)

---

**intersectSankey**

Plot Sankey diagram for intersecting sets

**Description**

Plot Sankey diagram for intersecting set through an array

**Usage**

intersectSankey(
  x,
  out.fig = NULL,
  color = NULL,
  step.names = c("Levels", "Variables", "Tasks"),
  fontSize = c(20, 13, 20),
  nodePadding = 11,
  nodeWidth = 5,
  margin = list(right = 180),
  ...
)

**Arguments**

- **x**: an array for constructing intersecting set
- **out.fig**: print the figure into "html", "pdf" or "png" file. Default is NULL with R graphics device
- **color**: a vector of colors corresponding to individual tasks
- **step.names**: names of the three dimensions of the array x, i.e. names of multiple levels, intermediate variables and tasks. Default is c("Levels", "Variables", "Tasks"). If step.names=NULL, it will not show the names
- **fontSize**: a value or vector of three values. If it is one value, it is the font size for all labels. But a vector of three values specifies the font size of the labels in the left, middle and right, respectively. Default is c(20, 10, 20)
- **nodePadding**: numeric essentially influences the width height
intersectSankey

nodeWidth numeric width of each node
margin an integer or a named list/vector of integers for the plot margins
... graphics parameters to be passed to sankeyNetwork() from R package networkD3

Value
An object of a D3 JavaScript intersecting Sankey diagram for visualising associations based on the input array.

Examples
# Data set 'cancers_genes_drugs' is an array with association scores between 56 genes (1st dimension), three cancer types (2nd dimension) and two drugs (3rd dimension)
data(cancers_genes_drugs, package = "EnrichIntersect")

intersectSankey(cancers_genes_drugs, step.names = c("Cancers", "Genes", "Drugs"))
Index

* datasets
  cancers_drug_groups, 2
  cancers_genes_drugs, 3

cancers_drug_groups, 2

enrichment, 3

intersectSankey, 5