Package 'DTSEA'

January 20, 2025

Type Package

Title Drug Target Set Enrichment Analysis

Version 0.0.3

Maintainer Junwei Han <hanjunwei1981@163.com>

Description It is a novel tool used to identify the candidate drugs against a particular disease based on the drug target set enrichment analysis. It assumes the most effective drugs are those with a closer affinity in the protein-protein interaction network to the specified disease. (See Gómez-Carballa et al. (2022) <doi:10.1016/j.envres.2022.112890> and Feng et al. (2022) <doi:10.7150/ijms.67815> for disease expression profiles; see Wishart et al. (2018) <doi:10.1093/nar/gkx1037> and Gaulton et al. (2017) <doi:10.1093/nar/gkw1074> for drug target information; see Kanehisa et al. (2021) <doi:10.1093/nar/gkaa970> for the details of KEGG database.)

License GPL (>= 2)

Depends R (>= 4.0.0)

Imports dplyr, fgsea, igraph, magrittr, tibble, tidyr, BiocParallel, stringr

Suggests knitr, rmarkdown

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.2.1

NeedsCompilation no

Author Junwei Han [aut, cre, cph], Yinchun Su [aut]

Repository CRAN

Date/Publication 2022-11-06 13:20:02 UTC

Contents

`DTSEA-package` *The Drug target set enrichment analysis (DTSEA)*

Description

The DTSEA implements a novel application to GSEA and extends the adoption of GSEA.

The Drug Target Set Enrichment Analysis (DTSEA) is a novel tool used to identify the most effective drug set against a particular disease based on the Gene Set Enrichment Analysis (GSEA).

The central hypothesis of DTSEA is that the targets of potential candidates for a specific disease (e.g., COVID-19) ought to be close to each other, or at least not so far away from the disease. The DTSEA algorithm determines whether a drug is potent for the chosen disease by the proximity between drug targets and the disease-related genes. Under the central hypothesis of DTSEA, the DTSEA consists of two main parts:

1. Evaluate the influence of the specific disease in the PPI network by the random walk with restart algorithm.
   To evaluate the influence, we compute the disease-node distance by using the random walk with restart (RwR) algorithm, then rank the nodes reversely.
2. Evaluate the drug-disease associations based on GSEA.
   The GSEA approach is adopted in this part to identify whether candidate drug targets are disease-related (top) or disease-unrelated (bottom) on the human PPI list. The specific disease gene list is normalized by the median and is set zero as the arbitrary cutoff point to classify the relations manually.

In this package, we provide the example data, which is a small set of data to demonstrate the usage and the main idea behind DTSEA. We provide some extra data files, the real data we used in the DTSEA paper. The supplementary package is now on the GitHub. Anyone can obtain this package by the example code.

Details

DTSEA

Examples

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

`calculate_p0` *Function to calculate the p0 vector used in Random Walk with Restart (RwR)*

Description

The function provides a reliable approach to generating a p0 vector.

Usage

```
calculate_p0(nodes, disease)
```

Arguments

`nodes`	The `nodes` variable can either accept the igraph object or the nodes vector.
`disease`	The `disease` variable must specify the disease-affected nodes in a short vector.

Value

The resulting p0 vector.

Examples

```
library(DTSEA)
library(dplyr)

# Load the data
data("example_disease_list", package = "DTSEA")
data("example_drug_target_list", package = "DTSEA")
data("example_ppi", package = "DTSEA")

# Compute the p0 vector
p0 <- calculate_p0(nodes = example_ppi, disease = example_disease_list)

# You can decrease the order of the p0 to get the most affected nodes.
p0 <- sort(p0, decreasing = TRUE) %>%
  names() %>%
  head(10)

# If you have obtained the supplemental data, then you can compute the p0
# in the real data set

# supp_data <- get_data(c("graph", "disease_related"))
```

```
# p0 <- calculate_p0(nodes = supp_data[["graph"]],
#                    disease = supp_data[["disease_related"]])
```

`cronbach.alpha` *Cronbach's alpha*

Description

Computes Cronbach's alpha

Usage

```
cronbach.alpha(data)
```

Arguments

`data`	A data frame or matrix contains n subjects * m raters.

Value

The Cronbach's alpha (unstandardized)

Examples

```
library(DTSEA)
library(tibble)

# Load the data
data <- tribble(~x, ~y, ~z, 1, 1, 2, 5, 6, 5, 7, 8, 4, 2, 3, 2, 8, 6, 5)

# Run Cronbach's alpha
cat(cronbach.alpha(data))
```

`DTSEA` *Main function of drug target set enrichment analysis (DTSEA)*

Description

The DTSEA function determines whether a drug is potent for a specific disease by the proximity between its targets and the disease-related genes.

Usage

```
DTSEA(
  network,
  disease,
  drugs,
  rwr.pt = 0,
  sampleSize = 101,
  minSize = 1,
  maxSize = Inf,
  nproc = 0,
  eps = 1e-50,
  nPermSimple = 5000,
  gseaParam = 1,
  verbose = TRUE
)
```

Arguments

network	The human protein-protein interactome network. It should be or be preconverted before being inputted in DTSEA.
disease	The disease-related nodes.
drugs	The drug-target long format dataframe. It includes at least columns with the drug_id and drug_target.
rwr.pt	The random walk p0 vector. Set it to 0 if you wish DTSEA automatically compute it, or you can provide your predetermined p0 vector.
sampleSize	The size of a randomly selected gene collection, where size = pathwaySize
minSize	Minimal set of a drug set to be tested.
maxSize	Maximal set of a drug set to be tested.
nproc	The CPU workers that fgsea would utilize.
eps	The boundary of calculating the p value.
nPermSimple	Number of permutations in the simple fgsea implementation for preliminary estimation of P-values.
gseaParam	GSEA parameter value, all gene-level statistics are raised to the power of 'gseaParam' before calculating of GSEA enrichment scores.
verbose	Show the messages

Value

The resulting dataframe consists of drug_id, pval, padj, log2err, ES, NES, size, and leadingEdge.

Examples

```
library(dplyr)
library(DTSEA)
```

```
# Load the data
data("example_disease_list", package = "DTSEA")
data("example_drug_target_list", package = "DTSEA")
data("example_ppi", package = "DTSEA")

# Run the DTSEA and sort the result dataframe by normalized enrichment scores
# (NES)
result <- DTSEA(
  network = example_ppi,
  disease = example_disease_list,
  drugs = example_drug_target_list,
  verbose = FALSE
) %>%
arrange(desc(NES))

# Or you can utilize the multi-core advantages by enable nproc parameters
# on non-Windows operating systems.
## Not run: result <- DTSEA(
         network = example_ppi,
         disease = example_disease_list,
         drugs = example_drug_target_list,
         nproc = 10, verbose = FALSE
)
## End(Not run)

# We can extract the significantly NES > 0 drug items.
result %>%
  filter(NES > 0 & pval < .05)
# Or we can draw the enrichment plot of the first predicted drug.
fgsea::plotEnrichment(
  pathway = example_drug_target_list %>%
    filter(drug_id == slice(result, 1)$drug_id) %>%
    pull(gene_target),
  stats = random.walk(network = example_ppi,
                      p0 = calculate_p0(nodes = example_ppi,
                                        disease = example_disease_list)
                      )
)

# If you have obtained the supplemental data, then you can do random walk
# with restart in the real data set

# supp_data <- get_data(c("graph", "disease_related", "example_ppi"))
# result <- DTSEA(network = supp_data[["graph"]],
#                 disease = supp_data[["disease_related"]],
#                 drugs = supp_data[["drug_targets"]],
#                 verbose = FALSE)
```

---

example_disease_list *An example vector of disease nodes*

Description

The list was integrated the significantly differentially expressed genes (DEGs) of GEO dataset GSE183071 and the work from Feng, Song, Guo, and et al.

Usage

```
example_disease_list
```

Format

An object of class `character` of length 63.

References

Gómez-Carballa A, Rivero-Calle I, Pardo-Seco J, Gómez-Rial J, Rivero-Velasco C, Rodríguez-Núñez N, Barbeito-Castiñeiras G, Pérez-Freixo H, Cebey-López M, Barral-Arca R, Rodriguez-Tenreiro C, Dacosta-Urbieta A, Bello X, Pischedda S, Currás-Tuala MJ, Viz-Lasheras S, Martinón-Torres F, Salas A; GEN-COVID study group. A multi-tissue study of immune gene expression profiling highlights the key role of the nasal epithelium in COVID-19 severity. Environ Res. 2022 Jul;210:112890. doi: 10.1016/j.envres.2022.112890. Epub 2022 Feb 22. PMID: 35202626; PMCID: PMC8861187.

Feng S, Song F, Guo W, Tan J, Zhang X, Qiao F, Guo J, Zhang L, Jia X. Potential Genes Associated with COVID-19 and Comorbidity. Int J Med Sci. 2022 Jan 24;19(2):402-415. doi: 10.7150/ijms.67815. PMID: 35165525; PMCID: PMC8795808.

Examples

```
library(DTSEA)

data("example_disease_list", package = "DTSEA")
```

`example_drug_target_list`

An example data frame of drug target lists

Description

Drug-target interactions were downloaded and integrated from DrugBank and ChEMBL.

Usage

```
example_drug_target_list
```

Format

A data frame with 970 rows and 3 variables:

- `drug_id`: the DrugBank ID
- `drug_name`: the name of each drug
- `gene_target`: the targets of drugs

References

Wishart DS, Feunang YD, Guo AC, Lo EJ, Marcu A, Grant JR, Sajed T, Johnson D, Li C, Sayeeda Z, Assempour N, Iynkkaran I, Liu Y, Maciejewski A, Gale N, Wilson A, Chin L, Cummings R, Le D, Pon A, Knox C, Wilson M. DrugBank 5.0: a major update to the DrugBank database for 2018. Nucleic Acids Res. 2018 Jan 4;46(D1):D1074-D1082. doi: 10.1093/nar/gkx1037. PMID: 29126136; PMCID: PMC5753335.

Gaulton A, Hersey A, Nowotka M, Bento AP, Chambers J, Mendez D, Mutowo P, Atkinson F, Bellis LJ, Cibrián-Uhalte E, Davies M, Dedman N, Karlsson A, Magariños MP, Overington JP, Papadatos G, Smit I, Leach AR. The ChEMBL database in 2017. Nucleic Acids Res. 2017 Jan 4;45(D1):D945-D954. doi: 10.1093/nar/gkw1074. Epub 2016 Nov 28. PMID: 27899562; PMCID: PMC5210557.

Examples

```
library(DTSEA)
data("example_drug_target_list", package = "DTSEA")
```

---

example_ppi *An example human gene functional interaction network object*

Description

We extracted the gene functional interaction network from multiple sources with experimental evidence and then integrated them.

Usage

```
example_ppi
```

Format

An igraph object

References

Kanehisa M, Furumichi M, Sato Y, Ishiguro-Watanabe M, Tanabe M. KEGG: integrating viruses and cellular organisms. Nucleic Acids Res. 2021 Jan 8;49 (D1):D545-D551. doi: 10.1093/nar/gkaa970. PMID: 33125081; PMCID: PMC7779016.

Examples

```
library(DTSEA)
data("example_ppi", package = "DTSEA")
```

`kendall.w` *Kendall's coefficient of concordance W*

Description

Computes the Kendall's coefficient of concordance.

Usage

```
kendall.w(raw, correct = TRUE)
```

Arguments

`raw`	A data frame or matrix contains n subjects * m raters.
`correct`	Logical. Indicates whether the W should be corrected for ties within raters.

Value

The resulting list consists of `title`, `kendall.w`, `chisq`, `df`, `pval`, `report`.

Examples

```
library(DTSEA)
library(tibble)

# Load the data
data <- tribble(~x, ~y, ~z, 1,1,2,  5,6,5,  7,8,4, 2,3,2, 8,6,5)

# Run Kendall's W
print(kendall.w(data)$report)
```

`random.walk` *Function to implement Random Walk with Restart (RwR) algorithm on the input graph*

Description

Function `random.walk` is supposed to implement the original Random Walk with Restart (RwR) on the input graph. If the seeds (i.e., a set of starting nodes) are given, it intends to calculate the affinity score of all nodes in the graph to the seeds.

Usage

```
random.walk(
  network,
  p0,
  edge_weight = FALSE,
  gamma = 0.7,
  threshold = 1e-10,
  pt.post.processing = "log",
  pt.align = "median",
  verbose = FALSE
)
```

Arguments

network	The input graph object. It should be either an igraph object or an edge list matrix / data frame.
p0	The starting vector on time t0.
edge_weight	Logical to indicate whether the input graph contains weight information.
gamma	The restart probability used for RwR. The gamma takes the value from **0** to **1**, controlling the probability that a node would go back to its starting node.
threshold	The threshold used for RwR. The threshold indicates the stabilization status, which is a stopping criterion of RwR.
pt.post.processing	The way to scale the pt vector. It can be 'none', 'zscore', and 'log'.
pt.align	The way to normalize the output pt vector. It can be **mean** to manually cut the up- and down-regulated genes, **median** to avoid the influence of the distribution shape, or **none** for no normalization.
verbose	Show the progress of the calculation.

Value

pt vector

Examples

```
library(DTSEA)

# Load the data
data("example_disease_list", package = "DTSEA")
data("example_drug_target_list", package = "DTSEA")
data("example_ppi", package = "DTSEA")

# Perform random walk
p0 <- calculate_p0(nodes = example_ppi, disease = example_disease_list)
pt <- random.walk(network = example_ppi, p0 = p0)

# Perform GSEA analysis
# ....
```

```
# If you have obtained the supplemental data, then you can do random walk
# with restart in the real data set

# supp_data <- get_data(c("graph", "disease_related", "example_ppi"))
# p0 <- calculate_p0(nodes = supp_data[["graph"]],
#                    disease = supp_data[["disease_related"]])
# pt <- random.walk(network = supp_data[["example_ppi"]],
#                   p0 = p0)
```

`random_graph` *A random graph for the computation of the separation measure*

Description

The random graph was retrieved from Menche et al (2015).

Usage

```
random_graph
```

Format

An igraph object

References

Menche J, Sharma A, Kitsak M, Ghiassian SD, Vidal M, Loscalzo J, Barabási AL. Disease networks. Uncovering disease-disease relationships through the incomplete interactome. Science. 2015 Feb 20;347(6224):1257601. doi: 10.1126/science.1257601. PMID: 25700523; PMCID: PMC4435741.

Examples

```
library(DTSEA)
data("random_graph", package = "DTSEA")
```

separation *A measure of network separation*

Description

Calculates the separation of two sets of nodes on a network. The metric is calculated as in Menche et al. (2015).

Usage

```
separation(graph, set_a, set_b)
```

Arguments

graph	The input graph object. It should be either an igraph object or an edge list matrix/data frame.
set_a	The first gene set
set_b	The second gene set

Value

The separation and distance measurement of the specified two modules.

Examples

```
library(DTSEA)

# Load the data
data("random_graph", package = "DTSEA")

# Compute the separation metric
separation <- separation(
  graph = random_graph,
  set_a = c("4", "6", "8", "13"),
  set_b = c("8", "9", "10", "15", "18")
)
cat(separation, "\n")
```

Index