Package ‘DTSEA’

October 18, 2022

Type Package
Title Drug Target Set Enrichment Analysis
Version 0.0.2
Maintainer Junwei Han <hanjunwei1981@163.com>
Description It is a novel tool used to identify the candidate drugs against a particular disease based on a 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
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 2022-10-18 07:25:32 UTC

R topics documented:

DTSEA-package

1
The Drug target set enrichment analysis (DTSEA)

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

Details
DTSEA

calculate_between

Description
Calculate between variance in network

Usage
calculate_between(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
A positive number
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)
### calculate_within

**Calculate within variance**

**Description**

No description

**Usage**

```r
calculate_within(graph, given_set)
```

**Arguments**

- `graph`: The input graph object. It should be either an igraph object or an edge list matrix/data frame.
- `given_set`: The first gene set

**Value**

A positive number

---

### cronbach.alpha

**Cronbach’s alpha**

**Description**

Computes Cronbach’s alpha

**Usage**

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

**Arguments**

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

**Value**

The Cronbach’s alpha (unstandardized)
DTSEA

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,
  eps = 1e-50,
  nPermSimple = 5000,
  gseaParam = 1,
  verbose = TRUE
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network</td>
<td>The human protein-protein interactome network. It should be or be preconverted before being inputted in DTSEA.</td>
</tr>
<tr>
<td>disease</td>
<td>The disease-related nodes.</td>
</tr>
<tr>
<td>drugs</td>
<td>The drug-target long format dataframe. It includes at least columns with the drug_id and drug_target.</td>
</tr>
<tr>
<td>rwr.pt</td>
<td>The random walk p0 vector. Set it to 0 if you wish DTSEA automatically compute it, or you can provide your predetermined p0 vector.</td>
</tr>
<tr>
<td>eps</td>
<td>The boundary of calculating the p value.</td>
</tr>
<tr>
<td>nPermSimple</td>
<td>Number of permutations in the simple fgsea implementation for preliminary estimation of P-values.</td>
</tr>
<tr>
<td>gseaParam</td>
<td>GSEA parameter value, all gene-level statistics are raised to the power of 'gsea-Param' before calculating of GSEA enrichment scores.</td>
</tr>
<tr>
<td>verbose</td>
<td>Show the messages</td>
</tr>
</tbody>
</table>
example_disease_list

Value


Examples

```r
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
) %>%
  arrange(desc(NES))

# 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)
)
)
```

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
example_drug_target_list

Format

An object of class character of length 63.

References


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

Examples

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

---

**example_ppi**

An example human protein-protein interactome graph object

Description

We extracted the protein-protein interactions from multiple biological pathways with experimental evidence and then integrated them from three different databases.

Usage

```r
example_ppi
```

Format

An igraph object

References


Examples

```r
library(DTSEA)
data("example_ppi", package = "DTSEA")
```
**kendall.w**

*Kendall’s coefficient of concordance W*

**Description**

Computes the Kendall’s coefficient of concordance.

**Usage**

```r
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**


**Examples**

```r
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.
random.walk

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 0.
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
# ....
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


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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>graph</td>
<td>The input graph object. It should be either an igraph object or an edge list matrix/data frame.</td>
</tr>
<tr>
<td>set_a</td>
<td>The first gene set</td>
</tr>
<tr>
<td>set_b</td>
<td>The second gene set</td>
</tr>
</tbody>
</table>
Value

The separation and distance measurement of the specified two modules.

Examples

```r
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

* datasets
  example_disease_list, 6
  example_drug_target_list, 7
  example_ppi, 8
  random_graph, 11

  calculate_between, 2
  calculate_p0, 3
  calculate_within, 4
  cronbach.alpha, 4

DTSEA, 5
DTsEA (DTSEA-package), 2
DTSEA-package, 2

  example_disease_list, 6
  example_drug_target_list, 7
  example_ppi, 8

  kendall.w, 9

random.walk, 9
random_graph, 11

separation, 11