Package ‘immcp’

October 13, 2022

Title Poly-Pharmacology Toolkit for Traditional Chinese Medicine Research

Version 1.0.3

Description Toolkit for Poly-pharmacology Research of Traditional Chinese Medicine. Based on the biological descriptors and drug-disease interaction networks, it can analyze the potential poly-pharmacological mechanisms of Traditional Chinese Medicine and be used for drug-repositioning in Traditional Chinese Medicine.

License GPL (>= 3)

URL `https://github.com/YuanlongHu/immcp`

BugReports `https://github.com/YuanlongHu/immcp/issues`

Depends igraph, R (>= 4.0.0)

Imports clusterProfiler, DOSE, dplyr, methods, magrittr, Matrix, openxlsx, org.Hs.eg.db, pbapply, proxyC, purrr, rlang (>= 1.0.2), stats, utils, visNetwork (>= 0.3.1), arules, ggplot2, ggheatmap, factoextra

Suggests knitr, rmarkdown

VignetteBuilder knitr

biocViews

Encoding UTF-8

RoxygenNote 7.1.2

LazyData true

NeedsCompilation no

Author Yuanlong Hu [aut, cre] (<`https://orcid.org/0000-0003-1933-7570`>)

Maintainer Yuanlong Hu <`huyuanlong1996@163.com`>

Repository CRAN

Date/Publication 2022-05-12 05:50:02 UTC

R topics documented:

`BasicData-class` *Class* `BasicData` *This class represents the basic input data.*

Description

Class `BasicData` This class represents the basic input data.

Slots

`drugnet` A directed graph

`vertices` Vertices of drug graph.

`diseasenet` Disease network.

`biomarker` Disease-related gene.

Author(s)

Yuanlong Hu

`BioDescr-class` *Class* `BioDescr` *This class represents the biological descriptor data.*

Description

Class `BioDescr` This class represents the biological descriptor data.

Slots

`drug_geneset` from drug to geneset.

`geneset_gene` from geneset to gene for each drug.

`anno` Geneset ID and description.

Author(s)

Yuanlong Hu

`CreateBasicData` *CreateBasicData*

Description

Create BasicData Object

Usage

```
CreateBasicData(..., diseasenet = NULL, biomarker = NULL)
```

Arguments

`...`	Drug graph from `PrepareData`.
`diseasenet`	A graph of Disease-related gene from `PrepareData`.
`biomarker`	Character vector, the vector of Disease-related gene.

Value

A BasicData object.

Author(s)

Yuanlong Hu

Examples

```
data(drugdemo)
drug_herb <- PrepareData(drugdemo$drug_herb, from = "drug", to="herb")
herb_compound <- PrepareData(drugdemo$herb_compound, from = "herb", to="compound")
compound_target <- PrepareData(drugdemo$compound_target, from = "compound", to="target")
disease <- PrepareData(drugdemo$disease, diseaseID = "disease",from = "target", to="target")
BasicData <- CreateBasicData(drug_herb, herb_compound, compound_target, diseasenet = disease)
```

`CreateDisDrugNet` *CreateDisDrugNet*

Description

Create Disease-Drug Network

Usage

```
CreateDisDrugNet(BasicData, drug, disease)
```

Arguments

`BasicData`	BasicData object.
`drug`	Character vector, the drug.
`disease`	Character vector, the disease.

Value

A igraph object.

Author(s)

Yuanlong Hu

Examples

```
data(drugdemo)
drug_herb <- PrepareData(drugdemo$drug_herb, from = "drug", to="herb")
herb_compound <- PrepareData(drugdemo$herb_compound, from = "herb", to="compound")
compound_target <- PrepareData(drugdemo$compound_target, from = "compound", to="target")
disease <- PrepareData(drugdemo$disease, diseaseID = "disease",from = "target", to="target")
BasicData <- CreateBasicData(drug_herb, herb_compound, compound_target, diseasenet = disease)
DisDrugNet <- CreateDisDrugNet(BasicData, drug = "Drug1", disease = "disease")
```

`diff_network_char` *diff_network_char*

Description

Calculate the difference of network characters in two network

Usage

```
diff_network_char(graph1, graph2, output_all = FALSE)
```

Arguments

`graph1`	A igraph object.
`graph2`	A igraph object.
`output_all`	FALSE

Value

A number vector.

Author(s)

Yuanlong Hu

`drugdemo` *Datasets Demo dataset*

Description

Datasets Demo dataset

`enrich_f` *enrich_f*

Description

Enrich Analysis

Usage

```
enrich_f(
  target_character,
  geneset = c("kegg", "mkegg", "go", "wp"),
  arguments = list(minGSSize = 5, maxGSSize = 500, pvalue = 0.05, qvalue = 0.1),
  out_dataframe = TRUE,
  to_ENTREZID = TRUE
)
```

Arguments

`target_character`	Charactor vector of gene.
`geneset`	Charactor vector, one of "kegg"(KEGG), "mkegg"(KEGG Module), "go"(GO-BP), and "wp"(WikiPathways); a data frame and list.
`arguments`	A list of the arguments of `clusterProfiler`, including `minGSSize`, `maxGSSize`, `pvalue`, and `qvalue`.
`out_dataframe`	Logical, whether to output data frame,defaults to `FALSE`.
`to_ENTREZID`	Logical, whether to translate to ENTREZID from SYMBOL, defaults to `TRUE`.

Value

data frame

Author(s)

Yuanlong Hu

`exportCytoscape` *Export an xlsx file to Cytoscape*

Description

Export an xlsx file to Cytoscape.

Usage

```
exportCytoscape(graph, file)
```

Arguments

`graph`	igraph object.
`file`	file

Value

A workbook object

Author(s)

Yuanlong Hu

`extr_biodescr` *Extract Biological descriptor*

Description

Extract Biological descriptor

Usage

```
extr_biodescr(
  BasicData,
  geneset = c("kegg", "mkegg", "go", "wp"),
  arguments = list(minGSSize = 5, maxGSSize = 500, pvalue = 0.05, qvalue = 0.1),
  ref_type = "drug",
  ref = NULL,
  to_ENTREZID = TRUE
)
```

Arguments

`BasicData`	BasicData object.
`geneset`	Charactor vector, one of "kegg"(KEGG), "mkegg"(KEGG Module), "go"(GO-BP), and "wp"(WikiPathways); a data frame and list.
`arguments`	A list of the arguments of `clusterProfiler`, including `minGSSize`, `maxGSSize`, `pvalue`, and `qvalue`.
`ref_type`	Charactor vector, one of "drug", "herb", "compound" or "target", defaults to "drug".
`ref`	Charactor vector, reference drug, herb, compound or target, defaults to `NULL`.
`to_ENTREZID`	Logical, whether to translate to ENTREZID from SYMBOL, defaults to TRUE.

Value

A BioDescr object.

Author(s)

Yuanlong Hu

`extr_biodescr,BasicData-method`
Extract Biological descriptor

Description

Extract Biological descriptor

Usage

```
## S4 method for signature 'BasicData'
extr_biodescr(
  BasicData,
  geneset = c("kegg", "mkegg", "go", "wp"),
  arguments = list(minGSSize = 5, maxGSSize = 500, pvalue = 0.05, qvalue = 0.1),
  ref_type = "drug",
  ref = NULL,
  to_ENTREZID = TRUE
)
```

Arguments

`BasicData`	BasicData object.
`geneset`	Charactor vector, one of "kegg"(KEGG), "mkegg"(KEGG Module), "go"(GO-BP), and "wp"(WikiPathways); a data frame and list.

`arguments`	A list of the arguments of `clusterProfiler`, including `minGSSize`, `maxGSSize`, `pvalue`, and `qvalue`.
`ref_type`	Charactor vector, one of "drug", "herb", "compound" or "target", defaults to "drug".
`ref`	Charactor vector, reference drug, herb, compound or target, defaults to `NULL`.
`to_ENTREZID`	Logical, whether to translate to ENTREZID from SYMBOL, defaults to TRUE.

Value

A BioDescr object.

Examples

```
## Not run:
data(drugdemo)
drug_herb <- PrepareData(drugdemo$drug_herb, from = "drug", to="herb")
herb_compound <- PrepareData(drugdemo$herb_compound, from = "herb", to="compound")
compound_target <- PrepareData(drugdemo$compound_target, from = "compound", to="target")
disease <- PrepareData(drugdemo$disease, diseaseID = "disease",from = "target", to="target")
BasicData <- CreateBasicData(drug_herb, herb_compound, compound_target, diseasenet = disease)
biodescr <- extr_biodescr(BasicData, geneset= "kegg")

## End(Not run)
```

`HerbResult-class` *Class* `HerbResult` *This class represents the biological descriptor data.*

Description

Class `HerbResult` This class represents the biological descriptor data.

Slots

`Drug_Herb`	Data frame, Drug-herb relationship.
`Herb_Herb`	Herb-herb association Rule Graph, it is a directed graph.

Author(s)

Yuanlong Hu

`natural_connectivity` *natural_connectivity*

Description

Calculate the natural connectivity

Usage

```
natural_connectivity(graph)
```

Arguments

`graph`	A igraph object.

Value

A numeric vector.

Author(s)

Yuanlong Hu

`network_char` *network_char*

Description

Calculate the network characters

Usage

```
network_char(graph, total_network = FALSE)
```

Arguments

`graph`	The graph.
`total_network`	Calculate for total network or each nodes.

Value

A number vector or data frame.

Author(s)

Yuanlong Hu

`network_node_ks` *network_node_ks*

Description

Kolmogorov-Smirnov tests for node characters between networks

Usage

```
network_node_ks(graph1, graph2, replicate = 1000)
```

Arguments

`graph1`	A igraph object.
`graph2`	A igraph object.
`replicate`	Number vector, the number of conduct bootstrapping sampling replications.

Value

A data frame

Author(s)

Yuanlong Hu

`plot_BioDescr` *Plot Biological descriptor*

Description

Plot Biological descriptor

Usage

```
plot_BioDescr(
  BioDescr,
  type = "heatmap",
  cluster_k = 2,
  colors = c("#2E9FDF", "#E7B800")
)
```

Arguments

`BioDescr`	BioDescr object.
`type`	one of "heatmap" and "clusterplot".
`cluster_k`	Number vector, number of cluster.
`colors`	vector of colors.

Value

Returns NULL, invisibly.

`plot_graph` *Plot Disease-Drug Network*

Description

Plot Disease-Drug Network

Usage

```
plot_graph(
  graph,
  drug,
  disease,
  Isolated = TRUE,
  vis = "visNetwork",
  color = c(drug = "#cca4e3", herb = "#ff461f", compound = "#ffc773", target =
    "#70f3ff"),
  width = 1,
  size = 20,
  ...
)

## S4 method for signature 'BasicData'
plot_graph(
  graph,
  drug,
  disease,
  Isolated = TRUE,
  vis = "visNetwork",
  color = c(drug = "#cca4e3", herb = "#ff461f", compound = "#ffc773", target =
    "#70f3ff"),
  width = 1,
  size = 20,
  ...
)

## S4 method for signature 'igraph'
plot_graph(
  graph,
  drug,
  disease,
  Isolated = TRUE,
  vis = "visNetwork",
```

```
    color = c(drug = "#cca4e3", herb = "#ff461f", compound = "#ffc773", target =
      "#70f3ff"),
    width = 1,
    size = 20,
    ...
  )

  ## S4 method for signature 'HerbResult'
  plot_graph(
    graph,
    drug,
    disease,
    Isolated = TRUE,
    vis = "visNetwork",
    color = c(drug = "#cca4e3", herb = "#ff461f", compound = "#ffc773", target =
      "#70f3ff"),
    width = 1,
    size = 20,
    ...
  )
```

Arguments

`graph`	graph.
`drug`	drug.
`disease`	disease.
`Isolated`	Whether to delect Isolated nodes.
`vis`	one of "igraph", "visNetwork" and "shiny".
`color`	Nodes Color
`width`	Edges width
`size`	Nodes size
`...`	Arguments

Value

Returns NULL, invisibly.

Author(s)

Yuanlong Hu

`PrepareData` *PrepareData*

Description

Prepare input format.

Usage

```
PrepareData(..., from, to, diseaseID, format = "single", sep)
```

Arguments

`...`	data frame, containing interaction information.
`from`	A charactor vector, containing "drug", "herb", "compound", or "target".
`to`	A character vector, containing "drug", "herb", "compound", or "target".
`diseaseID`	Charactor vector, diseaseID
`format`	one of "single" or "basket".
`sep`	Separator.

Value

A igraph object.

Author(s)

Yuanlong Hu

Examples

```
data(drugdemo)
drug_herb <- PrepareData(drugdemo$drug_herb, from = "drug", to="herb")
herb_compound <- PrepareData(drugdemo$herb_compound, from = "herb", to="compound")
compound_target <- PrepareData(drugdemo$compound_target, from = "compound", to="target")
disease <- PrepareData(drugdemo$disease, diseaseID = "disease",from = "target", to="target")
```

`read_gmt` *write_gmt*

Description

parse gmt file to a data.frame

Usage

```
read_gmt(gmtfile, out_dataframe = TRUE)
```

Arguments

`gmtfile`	A GMT file name or URL containing gene sets.
`out_dataframe`	TRUE or FALSE

Value

data.frame, list

Author(s)

Yuanlong Hu

`score_network` *score_network*

Description

Calculating differences in disease network characteristics before and after removal of drug targets

Usage

```
score_network(BasicData, n = 1000)
```

Arguments

`BasicData`	A BasicData object.
`n`	Number vector, the number of times random permutation sampling, default to 1000.

Value

A list.

Author(s)

Yuanlong Hu

Examples

```
data(drugdemo)
drug_herb <- PrepareData(drugdemo$drug_herb, from = "drug", to="herb")
herb_compound <- PrepareData(drugdemo$herb_compound, from = "herb", to="compound")
compound_target <- PrepareData(drugdemo$compound_target, from = "compound", to="target")
disease <- PrepareData(drugdemo$disease, diseaseID = "disease",from = "target", to="target")
BasicData <- CreateBasicData(drug_herb, herb_compound, compound_target, diseasenet = disease)
res <- score_network(BasicData, n = 100)
```

`score_rule` *Mining herb-herb associations with Apriori*

Description

Mine herb-herb association rules of prescription using the Apriori algorithm.

Usage

```
score_rule(BasicData, drug = NULL, support = 0.1, confidence = 0.8)
```

Arguments

`BasicData`	BasicData object.
`drug`	Charactor vector of drug names to analyze, default to `NULL`.
`support`	A numeric value for the minimal support of an item set, default to 0.1.
`confidence`	A numeric value for the minimal confidence of an item set, default to 0.8.

Value

A HerbResult object.

Author(s)

Yuanlong Hu

Examples

```
## Not run:
data(drugdemo)
drug_herb <- PrepareData(drugdemo$drug_herb, from = "drug", to="herb")
herb_compound <- PrepareData(drugdemo$herb_compound, from = "herb", to="compound")
compound_target <- PrepareData(drugdemo$compound_target, from = "compound", to="target")
disease <- PrepareData(drugdemo$disease, diseaseID = "disease",from = "target", to="target")
BasicData <- CreateBasicData(drug_herb, herb_compound, compound_target, diseasenet = disease)
```

```
res <- score_rule(BasicData, support = 0.1,confidence = 0.8)

## End(Not run)
```

`score_sim` *Calculating similarity between drug and disease*

Description

Calculating drug-disease similarity based on biological descriptors

Usage

```
score_sim(BioDescr, method = "jaccard", n = 1000)
```

Arguments

`BioDescr`	BioDescr object.
`method`	method to compute similarity, default "jaccard". See `proxyC::simil`.
`n`	number.

Value

A list.

Author(s)

Yuanlong Hu

Examples

```
## Not run:
data(drugdemo)
drug_herb <- PrepareData(drugdemo$drug_herb, from = "drug", to="herb")
herb_compound <- PrepareData(drugdemo$herb_compound, from = "herb", to="compound")
compound_target <- PrepareData(drugdemo$compound_target, from = "compound", to="target")
disease <- PrepareData(drugdemo$disease, diseaseID = "disease",from = "target", to="target")
BasicData <- CreateBasicData(drug_herb, herb_compound, compound_target, diseasenet = disease)
biodescr <- extr_biodescr(BasicData, geneset= "kegg")
res <- score_sim(biodescr, method="jaccard", n=1000)

## End(Not run)
```

`to_biodescr` *to_biodescr*

Description

Convert BioDescr object to a list of adjacency matrix

Usage

```
to_biodescr(BioDescr)
```

Arguments

`BioDescr`	A BioDescr object.

Value

A list.

Author(s)

Yuanlong Hu

Examples

```
## Not run:
  to_biodescr(BioDescr)

## End(Not run)
```

`to_df` *to_df*

Description

Convert list to data.frame

Usage

```
to_df(list)
```

Arguments

`list`	A list containing gene sets.

Value

A data frame.

Author(s)

Yuanlong Hu

Examples

```
## Not run:
  to_df(list)

## End(Not run)
```

`to_list` *to_list*

Description

Create a new list from a data.frame of drug target and disease biomarker as input

Usage

```
to_list(dataframe, input = "single", sep = ", ")
```

Arguments

`dataframe`	a data frame of 2 column with term/drug and gene
`input`	one of the single or basket
`sep`	When 'input' is 'basket'.

Value

list

Author(s)

Yuanlong Hu

Examples

```
## Not run:
  to_list(dataframe)

## End(Not run)
```

`write_gmt` *write_gmt*

Description

prints data frame to a gmt file

Usage

```
write_gmt(geneset, gmt_file)
```

Arguments

`geneset`	A data.frame of 2 column with term/drug and gene.
`gmt_file`	A character of gmt file name.

Value

gmt file

Author(s)

Yuanlong Hu

Index