Package 'WebGestaltR'

July 24, 2020

Type Package

Title Gene Set Analysis Toolkit WebGestaltR

Version 0.4.4

Date 2020-07-23

Description The web version WebGestalt <http://www.webgestalt.org> supports 12 organisms, 354 gene identifiers and 321,251 function categories. Users can upload the data and functional categories with their own gene identifiers. In addition to the Over-Representation Analysis, WebGestalt also supports Gene Set Enrichment Analysis and Network Topology Analysis. The user-friendly output report allows interactive and efficient exploration of enrichment results. The WebGestaltR package not only supports all above functions but also can be integrated into other pipeline or simultaneously analyze multiple gene lists.

License LGPL

URL https://github.com/bzhanglab/WebGestaltR

LazyLoad yes

Depends R (>= 3.3)

Imports methods, dplyr, doRNG, readr, parallel (>= 3.3.2), doParallel (>= 1.0.10), foreach (>= 1.4.0), jsonlite, httr, rlang, svglite, igraph, whisker, apcluster, Rcpp

NeedsCompilation yes

LinkingTo Rcpp

SystemRequirements C++11

RoxygenNote 7.0.2

Author Jing Wang [aut], Yuxing Liao [aut, cre], Eric Jaehnig [ctb], Zhiao Shi [ctb], Quanhu Sheng [ctb]

Maintainer Yuxing Liao <yuxingliao@gmail.com>

Repository CRAN

Date/Publication 2020-07-24 05:50:02 UTC

R topics documented:

`affinityPropagation` *Affinity Propagation*

Description

Use affinity propagation to cluster similar gene sets to reduce redundancy in report.

Usage

```
affinityPropagation(idsInSet, score)
```

Arguments

`idsInSet`	A list of set names and their member IDs.
`score`	A vector of addible scores with the same length used to assign input preference; higher score has larger weight, i.e. -logP.

Value

A list of `clusters` and `representatives` for each cluster.

clusters A list of character vectors of set IDs in each cluster.

representatives A character vector of representatives for each cluster.

Author(s)

Zhiao Shi, Yuxing Liao

formatCheck *Check Format and Read Data*

Description

Check Format and Read Data

Usage

```
formatCheck(dataType = "list", inputGeneFile = NULL, inputGene = NULL)
```

Arguments

dataType	Type of data, either list, rnk or gmt. Could be list, rnk or matrix for idToSymbol.
inputGeneFile	The data file to be mapped.
inputGene	Or the input could be given as an R object. GMT file should be read with readGmt.

Value

A list of data frame

goSlimSummary *GO Slim Summary*

Description

Outputs a brief summary of input genes based on GO Slim data.

Usage

```
goSlimSummary(
  organism = "hsapiens",
  geneList,
  outputFile,
  outputType = "pdf",
  isOutput = TRUE,
  cache = NULL,
  hostName = "http://www.webgestalt.org"
)
```

Arguments

organism	Currently, WebGestaltR supports 12 organisms. Users can use the function listOrganism to check available organisms. Users can also input others to perform the enrichment analysis for other organisms not supported by WebGestaltR. For other organisms, users need to provide the functional categories, interesting list and reference list (for ORA method). Because WebGestaltR does not perform the ID mapping for the other organisms, the above data should have the same ID type.
geneList	A list of input genes.
outputFile	Output file name.
outputType	File format of the plot: pdf, bmp or png.
isOutput	Boolean if a plot is save to outputFile.
cache	A directory to save data cache for reuse. Defaults to NULL and disabled.
hostName	The server URL for accessing data. Mostly for development purposes.

Value

A list of the summary result.

---

idMapping *ID Mapping*

Description

ID mapping utility with WebGestalt server.

Usage

```
idMapping(
  organism = "hsapiens",
  dataType = "list",
  inputGeneFile = NULL,
  inputGene = NULL,
  sourceIdType,
  targetIdType = NULL,
  collapseMethod = "mean",
  mappingOutput = FALSE,
  outputFileName = "",
  cache = NULL,
  hostName = "http://www.webgestalt.org/"
)

idToSymbol(
  organism = "hsapiens",
  dataType = "list",
```

```
  inputGeneFile = NULL,
  inputGene = NULL,
  sourceIdType = "ensembl_gene_id",
  collapseMethod = "mean",
  mappingOutput = FALSE,
  outputFileName = NULL,
  cache = NULL,
  hostName = "http://www.webgestalt.org/"
)
```

Arguments

organism	Currently, WebGestaltR supports 12 organisms. Users can use the function listOrganism to check available organisms. Users can also input others to perform the enrichment analysis for other organisms not supported by WebGestaltR. For other organisms, users need to provide the functional categories, interesting list and reference list (for ORA method). Because WebGestaltR does not perform the ID mapping for the other organisms, the above data should have the same ID type.
dataType	Type of data, either list, rnk or gmt. Could be list, rnk or matrix for idToSymbol.
inputGeneFile	The data file to be mapped.
inputGene	Or the input could be given as an R object. GMT file should be read with readGmt.
sourceIdType	The ID type of the data.
targetIdType	The ID type of the mapped data.
collapseMethod	The method to collapse duplicate IDs with scores. mean, median, min and max represent the mean, median, minimum and maximum of scores for the duplicate IDs.
mappingOutput	Boolean if the mapping output is written to file.
outputFileName	The output file name.
cache	A directory to save data cache for reuse. Defaults to NULL and disabled.
hostName	The server URL for accessing data. Mostly for development purposes.

Value

A list of mapped and unmapped IDs.

jaccardSim *Jaccard Similarity*

Description

Calculate Jaccard Similarity.

Usage

```
jaccardSim(idsInSet, score)
```

Arguments

idsInSet	A list of set names and their member IDs.
score	A vector of addible scores with the same length used to assign input preference; higher score has larger weight, i.e. -logP.

Value

A list of similarity matrix `sim.mat` and input preference vector `ip.vec`.

Author(s)

Zhiao Shi, Yuxing Liao

listArchiveUrl *List WebGestalt Servers*

Description

List available WebGestalt servers.

Usage

```
listArchiveUrl()
```

Value

A data frame of available servers.

listGeneSet *List Gene Sets*

Description

List available gene sets for the given organism on WebGestalt server.

Usage

```
listGeneSet(
  organism = "hsapiens",
  hostName = "http://www.webgestalt.org/",
  cache = NULL
)
```

Arguments

organism	Currently, WebGestaltR supports 12 organisms. Users can use the function `listOrganism` to check available organisms. Users can also input `others` to perform the enrichment analysis for other organisms not supported by WebGestaltR. For other organisms, users need to provide the functional categories, interesting list and reference list (for ORA method). Because WebGestaltR does not perform the ID mapping for the other organisms, the above data should have the same ID type.
hostName	The server URL for accessing data. Mostly for development purposes.
cache	A directory to save data cache for reuse. Defaults to `NULL` and disabled.

Value

A data frame of available gene sets.

listIdType *List ID Types*

Description

List supported ID types for the given organism on WebGestalt server.

Usage

```
listIdType(
  organism = "hsapiens",
  hostName = "http://www.webgestalt.org/",
  cache = NULL
)
```

Arguments

organism	Currently, WebGestaltR supports 12 organisms. Users can use the function `listOrganism` to check available organisms. Users can also input `others` to perform the enrichment analysis for other organisms not supported by WebGestaltR. For other organisms, users need to provide the functional categories, interesting list and reference list (for ORA method). Because WebGestaltR does not perform the ID mapping for the other organisms, the above data should have the same ID type.
hostName	The server URL for accessing data. Mostly for development purposes.
cache	A directory to save data cache for reuse. Defaults to `NULL` and disabled.

Value

A list of supported gene sets.

`listOrganism` *List Organisms*

Description

List supported organisms on WebGestalt server.

Usage

```
listOrganism(hostName = "http://www.webgestalt.org/", cache = NULL)
```

Arguments

hostName	The server URL for accessing data. Mostly for development purposes.
cache	A directory to save data cache for reuse. Defaults to `NULL` and disabled.

Value

A list of supported organisms.

listReferenceSet *List Reference Sets*

Description

List available reference sets for the given organism on WebGestalt server.

Usage

```
listReferenceSet(
  organism = "hsapiens",
  hostName = "http://www.webgestalt.org/",
  cache = NULL
)
```

Arguments

organism	Currently, WebGestaltR supports 12 organisms. Users can use the function listOrganism to check available organisms. Users can also input others to perform the enrichment analysis for other organisms not supported by WebGestaltR. For other organisms, users need to provide the functional categories, interesting list and reference list (for ORA method). Because WebGestaltR does not perform the ID mapping for the other organisms, the above data should have the same ID type.
hostName	The server URL for accessing data. Mostly for development purposes.
cache	A directory to save data cache for reuse. Defaults to NULL and disabled.

Value

A list of reference sets.

loadGeneSet *Load gene set data*

Description

Load gene set data

Usage

```
loadGeneSet(
  organism = "hsapiens",
  enrichDatabase = NULL,
  enrichDatabaseFile = NULL,
  enrichDatabaseType = NULL,
  enrichDatabaseDescriptionFile = NULL,
  cache = NULL,
  hostName = "http://www.webgestalt.org/"
)
```

Arguments

organism	Currently, WebGestaltR supports 12 organisms. Users can use the function `listOrganism` to check available organisms. Users can also input `others` to perform the enrichment analysis for other organisms not supported by WebGestaltR. For other organisms, users need to provide the functional categories, interesting list and reference list (for ORA method). Because WebGestaltR does not perform the ID mapping for the other organisms, the above data should have the same ID type.
enrichDatabase	The functional categories for the enrichment analysis. Users can use the function `listGeneSet` to check the available functional databases for the selected organism. Multiple databases in a vector are supported for ORA and GSEA.
enrichDatabaseFile	Users can provide one or more GMT files as the functional category for enrichment analysis. The extension of the file should be `gmt` and the first column of the file is the category ID, the second one is the external link for the category. Genes annotated to the category are from the third column. All columns are separated by tabs. The GMT files will be combined with `enrichDatabase`.
enrichDatabaseType	The ID type of the genes in the `enrichDatabaseFile`. If users set `organism` as `others`, users do not need to set this ID type because WebGestaltR will not perform ID mapping for other organisms. The supported ID types of WebGestaltR for the selected organism can be found by the function `listIdType`.
enrichDatabaseDescriptionFile	Users can also provide description files for the custom `enrichDatabaseFile`. The extension of the description file should be `des`. The description file contains two columns: the first column is the category ID that should be exactly the same as the category ID in the custom `enrichDatabaseFile` and the second column is the description of the category. All columns are separated by tabs.
cache	A directory to save data cache for reuse. Defaults to `NULL` and disabled.
hostName	The server URL for accessing data. Mostly for development purposes.

Value

A list of `geneSet`, `geneSetDes`, `geneSetDag`, `geneSetNet`, `standardId`.

geneSet Gene set: A data frame with columns of "geneSet", "description", "genes"

geneSetDes Description: A data frame with columns of two columns of gene set ID and description

geneSetDag DAG: A edge list data frame of two columns of parent and child. Or a list of data frames if multilple databases are given.

geneSetNet Network: A edge list data frame of two columns connecting nodes. Or a list of data frames if multilple databases are given.

standardId The standard ID of the gene set

`prepareGseaInput` *Prepare input for standard GSEA*

Description

A helper to read files for performing standard GSEA.

Usage

```
prepareGseaInput(rankFile, gmtFile)
```

Arguments

`rankFile`	Path of the rnk file
`gmtFile`	Path of the GMT file

Value

a data frame to be used in `swGsea`

`prepareInputMatrixGsea` *Prepare Input Matrix for GSEA*

Description

Prepare Input Matrix for GSEA

Usage

```
prepareInputMatrixGsea(rank, gmt)
```

Arguments

`rank`	A 2 column Data Frame of gene and score
`gmt`	3 column Data Frame of geneSet, description, and gene

Value

A matrix used for input to `swGsea`.

readGmt *Read GMT File*

Description

Read GMT File

Usage

```
readGmt(gmtFile, cache = NULL)
```

Arguments

gmtFile	The file path or URL of the GMT file.
cache	A directory to save data cache for reuse. Defaults to NULL and disabled.

Value

A data frame with columns of "geneSet", "description", "gene".

swGsea *Site Weighted Gene Set Enrichment Analysis*

Description

Performs site weighted gene set enrichment analysis or standard GSEA when likelihood/weight columns in input_df are 1 or 0, p=1, q=1 and thresh_type="val".

Usage

```
swGsea(
  input_df,
  thresh_type = "percentile",
  thresh = 0.9,
  thresh_action = "exclude",
  min_set_size = 10,
  max_set_size = 500,
  max_score = "max",
  min_score = "min",
  psuedocount = 0.001,
  perms = 1000,
  p = 1,
  q = 1,
  nThreads = 1,
  rng_seed = 1,
  fork = FALSE
)
```

Arguments

input_df	A data frame in which first column is name of item of interest (gene, protein, phosphosite, etc.), the second is the correlation of that item of interest with the phenotype (typically log ratio of expression for phenotype vs. normal), and the remaining columns are the scores for the likelihood that the item belongs in each set (one column per set).
thresh_type	The type of thresh. Use 'percentile' to include all scores over that percentile given in thresh (i.e., 0.9 would be all items in 90th percentile, or top 10 percent); 'list' to include a list of set lists where the set lists are in the same order as the corresponding set columns in the input_df; 'val' to apply a single threshold value to all sets; or 'values' to use a vector of unique cutoffs for each set (needs to be in the same order as the sets are specified in the columns of input_df")
thresh	Depends on thresh_type. A list of lists of the items in each set (with same names as colnames of the scores); a numeric vector of threshold scores for each set (in the same order as the colnames of the scores in the input_df), or a single percentile value between 0 and 1 (i.e., if thresh=0.9, the 90th percentile of the score or the highest scoring 10 of of the items are included in the set for each scoring regimen) (thresh ="all" is not supported at this time, as it doesn't result in a Kolgorov-Smirnoff statistic; this may be worked in as an alternate scoring method later on).
thresh_action	Either "include", "exclude (default)", or "adjust"; this specifies how to treat each set if it doesn't contain a minimum number of items or contains all of the items; this option cannot be used with predefined lists of items in sets (if the number of items in a given set doesn't meet requirements, that set will be skipped).
min_set_size, max_set_size	The minimum/maximum number of items each set needs for the analysis to proceed.
max_score, min_score	A optional numeric vector of minimum/maximum boundaries to clip scores for each set.
psuedocount	Psuedocount (pc) is used for rescaling set scores: (score -min_score + pc)/(max_score -min_score +pc); this is needed to prevent division by 0 if max_score==min_score (in this case, all scores for items in set will be 1, which is equivalent to standard GSEA); it also allows users to adjust weights for scores that are close to the minimum for the scores in the set (unless min_score==max_score): as psuedocount value approaches 0, scaled minimum scores also approach 0; as psuedocount approaches infinity, scaled minimum scores approach the scaled maximum scores (which equal 1); this value must be larger than 0.
perms	The number of permutations.
p	The exponential scaling factor of the phenotype score (second column in input_df).
q	The exponential scaling factor of the likelihood score (weights).
nThreads	The number of threads to use in calculating permutaions.
rng_seed	Random seed.
fork	A boolean. Whether pass "fork" to type parameter of makeCluster on Unix-like machines.

Details

The formula for weighting is as follows

$$\frac{s_j^q |r_j|^p}{\sum s^q |r|^p}$$

Where r is log ratio score, s is likelihood score, j is the index of the gene.

Value

A list of `Enrichment_Results`, `Items_in_Set` and `Running_Sums`.

Enrichment_Results A data frame with row names of gene set and columns of "ES", "NES", "p_val", "fdr".

Items_in_Set A list of one-column data frames. Describes genes and their ranks in each set.

Running_Sums Running sum scores along genes sorted by ranked scores, with gene sets as columns.

Author(s)

Eric Jaehnig

`WebGestaltR` *WebGestaltR: The R interface for enrichment analysis with WebGestalt.*

Description

Main function for enrichment analysis

Usage

```
WebGestaltR(
  enrichMethod = "ORA",
  organism = "hsapiens",
  enrichDatabase = NULL,
  enrichDatabaseFile = NULL,
  enrichDatabaseType = NULL,
  enrichDatabaseDescriptionFile = NULL,
  interestGeneFile = NULL,
  interestGene = NULL,
  interestGeneType = NULL,
  collapseMethod = "mean",
  referenceGeneFile = NULL,
  referenceGene = NULL,
  referenceGeneType = NULL,
  referenceSet = NULL,
  minNum = 10,
```

```
  maxNum = 500,
  sigMethod = "fdr",
  fdrMethod = "BH",
  fdrThr = 0.05,
  topThr = 10,
  reportNum = 20,
  perNum = 1000,
  gseaP = 1,
  isOutput = TRUE,
  outputDirectory = getwd(),
  projectName = NULL,
  dagColor = "continuous",
  saveRawGseaResult = FALSE,
  gseaPlotFormat = c("png", "svg"),
  setCoverNum = 10,
  networkConstructionMethod = NULL,
  neighborNum = 10,
  highlightType = "Seeds",
  highlightSeedNum = 10,
  nThreads = 1,
  cache = NULL,
  hostName = "http://www.webgestalt.org/",
  ...
)

WebGestaltRBatch(
  interestGeneFolder = NULL,
  enrichMethod = "ORA",
  isParallel = FALSE,
  nThreads = 3,
  ...
)
```

Arguments

`enrichMethod`	Enrichment methods: `ORA`, `GSEA` or `NTA`.
`organism`	Currently, WebGestaltR supports 12 organisms. Users can use the function `listOrganism` to check available organisms. Users can also input `others` to perform the enrichment analysis for other organisms not supported by WebGestaltR. For other organisms, users need to provide the functional categories, interesting list and reference list (for ORA method). Because WebGestaltR does not perform the ID mapping for the other organisms, the above data should have the same ID type.
`enrichDatabase`	The functional categories for the enrichment analysis. Users can use the function `listGeneSet` to check the available functional databases for the selected organism. Multiple databases in a vector are supported for ORA and GSEA.
`enrichDatabaseFile`	Users can provide one or more GMT files as the functional category for enrich-

ment analysis. The extension of the file should be gmt and the first column of the file is the category ID, the second one is the external link for the category. Genes annotated to the category are from the third column. All columns are separated by tabs. The GMT files will be combined with enrichDatabase.

enrichDatabaseType
: The ID type of the genes in the enrichDatabaseFile. If users set organism as others, users do not need to set this ID type because WebGestaltR will not perform ID mapping for other organisms. The supported ID types of WebGestaltR for the selected organism can be found by the function listIdType.

enrichDatabaseDescriptionFile
: Users can also provide description files for the custom enrichDatabaseFile. The extension of the description file should be des. The description file contains two columns: the first column is the category ID that should be exactly the same as the category ID in the custom enrichDatabaseFile and the second column is the description of the category. All columns are separated by tabs.

interestGeneFile
: If enrichMethod is ORA or NTA, the extension of the interestGeneFile should be txt and the file can only contain one column: the interesting gene list. If enrichMethod is GSEA, the extension of the interestGeneFile should be rnk and the file should contain two columns separated by tab: the gene list and the corresponding scores.

interestGene
: Users can also use an R object as the input. If enrichMethod is ORA or NTA, interestGene should be an R vector object containing the interesting gene list. If enrichMethod is GSEA, interestGene should be an R data.frame object containing two columns: the gene list and the corresponding scores.

interestGeneType
: The ID type of the interesting gene list. The supported ID types of WebGestaltR for the selected organism can be found by the function listIdType. If the organism is others, users do not need to set this parameter.

collapseMethod
: The method to collapse duplicate IDs with scores. mean, median, min and max represent the mean, median, minimum and maximum of scores for the duplicate IDs.

referenceGeneFile
: For the ORA method, the users need to upload the reference gene list. The extension of the referenceGeneFile should be txt and the file can only contain one column: the reference gene list.

referenceGene
: For the ORA method, users can also use an R object as the reference gene list. referenceGene should be an R vector object containing the reference gene list.

referenceGeneType
: The ID type of the reference gene list. The supported ID types of WebGestaltR for the selected organism can be found by the function listIdType. If the organism is others, users do not need to set this parameter.

referenceSet
: Users can directly select the reference set from existing platforms in WebGestaltR and do not need to provide the reference set through referenceGeneFile. All existing platforms supported in WebGestaltR can be found by the function listReferenceSet.

If referenceGeneFile and refereneceGene are NULL, WebGestaltR will use the referenceSet as the reference gene set. Otherwise, WebGestaltR will use the user supplied reference set for enrichment analysis.

minNum — WebGestaltR will exclude the categories with the number of annotated genes less than minNum for enrichment analysis. The default is 10.

maxNum — WebGestaltR will exclude the categories with the number of annotated genes larger than maxNum for enrichment analysis. The default is 500.

sigMethod — Two methods of significance are available in WebGestaltR: fdr and top. fdr means the enriched categories are identified based on the FDR and top means all categories are ranked based on FDR and then select top categories as the enriched categories. The default is fdr.

fdrMethod — For the ORA method, WebGestaltR supports five FDR methods: holm, hochberg, hommel, bonferroni, BH and BY. The default is BH.

fdrThr — The significant threshold for the fdr method. The default is 0.05.

topThr — The threshold for the top method. The default is 10.

reportNum — The number of enriched categories visualized in the final report. The default is 20. A larger reportNum may be slow to render in the report.

perNum — The number of permutations for the GSEA method. The default is 1000.

gseaP — The exponential scaling factor of the phenotype score. The default is 1. When p=0, ES reduces to standard K-S statistics (See original paper for more details).

isOutput — If isOutput is TRUE, WebGestaltR will create a folder named by the projectName and save the results in the folder. Otherwise, WebGestaltR will only return an R data.frame object containing the enrichment results. If hundreds of gene list need to be analyzed simultaneously, it is better to set isOutput to FALSE. The default is TRUE.

outputDirectory — The output directory for the results.

projectName — The name of the project. If projectName is NULL, WebGestaltR will use time stamp as the project name.

dagColor — If dagColor is binary, the significant terms in the DAG structure will be colored by steel blue for ORA method or steel blue (positive related) and dark orange (negative related) for GSEA method. If dagColor is continous, the significant terms in the DAG structure will be colored by the color gradient based on corresponding FDRs.

saveRawGseaResult — Whether the raw result from GSEA is saved as a RDS file, which can be used for plotting. Defaults to FALSE. The list includes

Enrichment_Results A data frame of GSEA results with statistics

Running_Sums A matrix of running sum of scores for each gene set

Items_in_Set A list with ranks of genes for each gene set

gseaPlotFormat — The graphic format of GSEA enrichment plots. Either svg, png, or c("png","svg") (default).

setCoverNum — The number of expected gene sets after set cover to reduce redundancy. It could get fewer sets if the coverage reaches 100%. The default is 10.

`networkConstructionMethod`
: Netowrk construction method for NTA. Either `Network_Retrieval_Prioritization` or `Network_Expansion`. Network Retrieval & Prioritization first uses random walk analysis to calculate random walk probabilities for the input seeds, then identifies the relationships among the seeds in the selected network and returns a retrieval sub-network. The seeds with the top random walk probabilities are highlighted in the sub-network. Network Expansion first uses random walk analysis to rank all genes in the selected network based on their network proximity to the input seeds and then return an expanded sub-network in which nodes are the input seeds and their top ranking neighbors and edges represent their relationships.

`neighborNum`
: The number of neighbors to include in NTA Network Expansion method.

`highlightType`
: The type of nodes to highlight in the NTA Network Expansion method, either `Seeds` or `Neighbors`.

`highlightSeedNum`
: The number of top input seeds to highlight in NTA Network Retrieval & Prioritizaiton method.

`nThreads`
: The number of cores to use for GSEA and set cover, and in batch function.

`cache`
: A directory to save data cache for reuse. Defaults to `NULL` and disabled.

`hostName`
: The server URL for accessing data. Mostly for development purposes.

`...`
: In batch function, passes parameters to WebGestaltR function. Also handles backward compatibility for some parameters in old versions.

`interestGeneFolder`
: Run WebGestaltR for gene list files in the folder.

`isParallel`
: If jobs are run parallelly in the batch.

Details

WebGestaltR function can perform three enrichment analyses: ORA (Over-Representation Analysis) and GSEA (Gene Set Enrichment Analysis).and NTA (Network Topology Analysis). Based on the user-uploaded gene list or gene list with scores, WebGestaltR function will first map the gene list to the entrez gene ids and then summarize the gene list based on the GO (Gene Ontology) Slim. After performing the enrichment analysis, WebGestaltR function also returns a user-friendly HTML report containing GO Slim summary and the enrichment analysis result. If functional categories have DAG (directed acyclic graph) structure or genes in the functional categories have network structure, those relationship can also be visualized in the report.

Value

The WebGestaltR function returns a data frame containing the enrichment analysis result and also outputs an user-friendly HTML report if `isOutput` is `TRUE`. The columns in the data frame depend on the `enrichMethod` and they are the following:

geneSet ID of the gene set.

description Description of the gene set if available.

link Link to the data source.

size The number of genes in the set after filtering by `minNum` and `maxNum`.

overlap The number of mapped input genes that are annotated in the gene set.

expect Expected number of input genes that are annotated in the gene set.

enrichmentRatio Enrichment ratio, overlap / expect.

enrichmentScore Enrichment score, the maximum running sum of scores for the ranked list.

normalizedEnrichmentScore Normalized enrichment score, normalized against the average enrichment score of all permutations.

leadingEdgeNum Number of genes/phosphosites in the leading edge.

pValue P-value from hypergeometric test for ORA. For GSEA, please refer to its original publication or online at https://software.broadinstitute.org/gsea/doc/GSEAUserGuideTEXT.htm.

FDR Corrected P-value for mulilple testing with `fdrMethod` for ORA.

overlapId The gene/phosphosite IDs of `overlap` for ORA (entrez gene IDs or phosphosite sequence).

leadingEdgeId Genes/phosphosites in the leading edge in entrez gene ID or phosphosite sequence.

userId The gene/phosphosite IDs of `overlap` for ORA or `leadingEdgeId` for GSEA in User input IDs.

plotPath Path of the GSEA enrichment plot.

database Name of the source database if multiple enrichment databases are given.

goId In NTA, like `geneSet`, the enriched GO terms of genes in the returned subnetwork.

interestGene In NTA, the gene IDs in the subnetwork with 0/1 annotations indicating if it is from user input.

The WebGestaltRBatch function returns a list of enrichment results.

Examples

```
## Not run:
####### ORA example #########
geneFile <- system.file("extdata", "interestingGenes.txt", package="WebGestaltR")
refFile <- system.file("extdata", "referenceGenes.txt", package="WebGestaltR")
outputDirectory <- getwd()
enrichResult <- WebGestaltR(enrichMethod="ORA", organism="hsapiens",
  enrichDatabase="pathway_KEGG", interestGeneFile=geneFile,
  interestGeneType="genesymbol", referenceGeneFile=refFile,
  referenceGeneType="genesymbol", isOutput=TRUE,
  outputDirectory=outputDirectory, projectName=NULL)

####### GSEA example #########
rankFile <- system.file("extdata", "GeneRankList.rnk", package="WebGestaltR")
outputDirectory <- getwd()
enrichResult <- WebGestaltR(enrichMethod="GSEA", organism="hsapiens",
  enrichDatabase="pathway_KEGG", interestGeneFile=rankFile,
  interestGeneType="genesymbol", sigMethod="top", topThr=10, minNum=5,
  outputDirectory=outputDirectory)
```

```
####### NTA example #########
enrichResult <- WebGestaltR(enrichMethod="NTA", organism="hsapiens",
  enrichDatabase="network_PPI_BIOGRID", interestGeneFile=geneFile,
  interestGeneType="genesymbol", sigMethod="top", topThr=10,
  outputDirectory=getwd(), highlightSeedNum=10,
  networkConstructionMethod="Network_Retrieval_Prioritization")

## End(Not run)
```

`weightedSetCover` *Weighted Set Cover*

Description

Size constrained weighted set cover problem to find top N sets while maximizing the coverage of all elements.

Usage

```
weightedSetCover(idsInSet, costs, topN, nThreads = 4)
```

Arguments

`idsInSet`	A list of set names and their member IDs.
`costs`	A vector of the same length to add weights for penalty, i.e. 1/-logP.
`topN`	The number of sets (or less when it completes early) to return.
`nThreads`	The number of processes to use. In Windows, it fallbacks to 1.

Value

A list of `topSets` and `coverage`.

topSets A list of set IDs.

coverage The percentage of IDs covered in the top sets.

Author(s)

Zhiao Shi, Yuxing Liao

Index