

Package: ExpHunterSuite (via r-universe)

October 4, 2026

Title Package for the Comprehensive Analysis of Transcriptomic Data

Version 1.20.0

Author James Perkins

Maintainer James Perkins <jimrperkins@gmail.com>

Description The ExpHunterSuite implements a comprehensive protocol for the analysis of transcriptional data using established R packages and combines their results. It covers key steps including differential expression analysis, co-expression analysis, and functional enrichment for RNA-seq data. The package provides interactive R functions as well as command-line scripts that wrap these functions.

License MIT + file LICENSE

Depends R (>= 4.1.0)

Imports AnnotationDbi, annotatr, Biobase, BiocParallel, clusterProfiler, data.table, DESeq2, diffcoexp, DOSE, dplyr, DT, edgeR, enrichit, enrichplot, FactoInvestigate, FactoMineR, fastcluster, GenomicFeatures, GenomicRanges, ggplot2, ggrepel, ggridges, GO.db, graphics, grDevices, heatmaply, knitr, limma, magrittr, Matrix, matrixStats, miRBaseVersions.db, MKinfer, NOISeq, plyr, ReactomePA, rlang, rmarkdown, ROCR, stats, stringr, SummarizedExperiment, tidyr, topGO, tximport, utils, txdbmaker, WGCNA

Suggests BiocGenerics, biomaRt, EnhancedVolcano, ggupset, naivebayes, optparse, org.Hs.eg.db, org.Mm.eg.db, PerformanceAnalytics, reshape2, testthat (>= 3.0.0)

VignetteBuilder knitr

biocViews GeneExpressionWorkflow

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Config/pak/sysreqs libcairo2-dev cmake libfontconfig1-dev libfreetype6-dev libglpk-dev libgmp3-dev make libmagick++-dev gsfonts libbz2-dev libicu-dev liblzma-dev libpng-dev libuv1-dev libxml2-dev libssl-dev pari-gp libx11-dev xz-utils zlib1g-dev

Repository Bioconductor 3.23

Date/Publication 2026-04-28 12:44:49 UTC

RemoteUrl <https://github.com/bioc/ExpHunterSuite>

RemoteRef RELEASE_3_23

RemoteSha 1960a2c9f7f2ae5f261065c1eb3b4229e4e7649c

Contents

acc	3
deg_h_output	3
deg_synth	4
download_latest_kegg_db	5
extract_from_cm	5
fcfunc	6
fmeasure	6
get_kegg_db_path	7
get_org_db	7
get_organism_table	8
get_stats	8
get_stats_from_cm	9
get_unexported_function	9
ht2logFCPlot	10
install_DEgenes_hunter	11
load_and_parse_gmt	11
load_synth_dataset	12
load_WGCNA_results	12
main_clusters_to_enrichment	13
main_degenes_Hunter	14
main_functional_hunter	16
multienricher_gsea	18
multienricher_oracle	19
multienricher_topGO	20
parallel_list	21
parse_pseudocounts	22
ppv	22
recall	23
rtable2measures	23
scale_data_matrix	24
scale_range	25
spc	25
STC	26
target	27
target_generation	27

<code>acc</code>	3
<code>toc</code>	28
<code>translate_from_table</code>	28
<code>trycatch_pairwise_termsim</code>	29
<code>write_clusters_to_enrichment</code>	29
<code>write_df_list_as_tables</code>	31
<code>write_enrich_files</code>	31
<code>write_expression_report</code>	32
<code>write_functional_report</code>	33
Index	35

<code>acc</code>	<i>Calculates accuracy from a measures dataframe</i>
------------------	--

Description

Calculates accuracy from a measures dataframe

Usage

`acc(df)`

Arguments

`df` dataframe with measures

Value

calculated metric

Examples

```
df <- data.frame(TP = 4, FP = 3, TN = 2, FN = 1)
acc(df)
```

<code>degh_output</code>	<i>Output data from the main_degenes_Hunter function</i>
--------------------------	--

Description

Output data from the main_degenes_Hunter function

Usage

`data(degh_output)`

Format

An output object from the main_degenes_Hunter function.

Examples

```
data(degh_output)
```

deg synth	<i>Main function to generate synthetic data using an specific exponential distribution for logFC</i>
-----------	--

Description

Main function to generate synthetic data using an specific exponential distribution for logFC

Usage

```
deg synth(
  outfile = NULL,
  inputfile = NULL,
  replicates = 3,
  ngenes = 20000,
  DEGs_proportion = 0.2,
  FC_min = 1.3,
  FC_max = 3,
  P_up = 1,
  group = NULL
)
```

Arguments

outfile	output file
inputfile	input file
replicates	number of replicates
ngenes	number of genes
DEGs_proportion	numeric (0,1) proportion of DEG genes to be simulated
FC_min	minimum Fold-Change
FC_max	maximum Fold-Change
P_up	proportion of up-regulated genes
group	optional group identifiers separated by commas

Value

synthetic data generated

Examples

```
synthetic_dataset <- degsynth()  
# Returns simulated count dataset and indication of which are DE/not DE
```

```
download_latest_kegg_db  
      download kegg db for a given organism
```

Description

download kegg db for a given organism

Usage

```
download_latest_kegg_db(current_organism_info, file)
```

Arguments

current_organism_info	
	organism info for which to download the file
file	where to save the file

extract_from_cm	<i>Extract value from confusion matrix</i>
-----------------	--

Description

Extract value from confusion matrix

Usage

```
extract_from_cm(cm, real, pred, default = 0)
```

Arguments

cm	confusion matrix
real	slot name
pred	slot name
default	to return if value is not available

Value

value required or default if it is not available

fcfunc	<i>Custom function to generate exponentially degraded foldchange range</i>
--------	--

Description

Custom function to generate exponentially degraded foldchange range

Usage

```
fcfunc(means, fcmín = 1.4, fcmax = 3, meanlog = 1, sdlog = 0.8)
```

Arguments

means	means vector
fcmín	minimum foldchange
fcmax	maximum foldchange
meanlog	param of rlnorm function
sdlog	param of rlnorm function

Value

a vector of foldchanges to be applied

fmeasure	<i>Calculates F measure from a measures dataframe</i>
----------	---

Description

Calculates F measure from a measures dataframe

Usage

```
fmeasure(df)
```

Arguments

df	dataframe with measures
----	-------------------------

Value

calculated metric

Examples

```
df <- data.frame(TP = 4, FP = 3, TN = 2, FN = 1)
df$ppv <- ppv(df)
df$recall <- recall(df)
fmeasure(df)
```

get_kegg_db_path	<i>get path for KEGG db for downloading/usage</i>
------------------	---

Description

get path for KEGG db for downloading/usage

Usage

```
get_kegg_db_path(kegg_data_file, current_organism_info)
```

Arguments

kegg_data_file file path. if null, inst/kegg_data_files will be used
current_organism_info
information for the organism, including KEGG code

get_org_db	<i>Obtain an org_db object from a current_organism_info, obtained from subsetting the results of get_organism_table()</i>
------------	---

Description

Obtain an org_db object from a current_organism_info, obtained from subsetting the results of
get_organism_table()

Usage

```
get_org_db(current_organism_info)
```

Arguments

current_organism_info
row from output of get_organism_table()

Value

org_db

get_organism_table	<i>Table with information about all organism available</i>
--------------------	--

Description

Table with information about all organism available

Usage

```
get_organism_table(file = NULL)
```

Arguments

file to be loaded. If none given, internal organism table loaded

Value

organism table

Examples

```
ot <- get_organism_table()
```

get_stats	<i>Function to calculate full set of measures from a confusion data frame</i>
-----------	---

Description

Function to calculate full set of measures from a confusion data frame

Usage

```
get_stats(df)
```

Arguments

df confusion dataframe

Value

a data frame with several measures

Examples

```
df <- data.frame(TP = 4, FP = 3, TN = 2, FN = 1)
stats <- get_stats(df)
```

get_stats_from_cm	<i>Obtain full set of measures from a confusion matrix</i>
-------------------	--

Description

Obtain full set of measures from a confusion matrix

Usage

```
get_stats_from_cm(cm)
```

Arguments

cm	confusion matrix
----	------------------

Value

a data frame with several measures

Examples

```
cm <- matrix(c(4,3,2,1), ncol = 2)
rownames(cm) <- c(TRUE,FALSE)
colnames(cm) <- c(TRUE,FALSE)
stats <- get_stats_from_cm(cm)
```

get_unexported_function	<i>Method used to import hidden functions legally for CRAN check</i>
-------------------------	--

Description

Method used to import hidden functions legally for CRAN check

Usage

```
get_unexported_function(pkg, fun)
```

Arguments

pkg	package
fun	function to be loaded

Value

function required

ht2logFCPlot	<i>Function to generate scatter plot for each gene in a hunter table and show logFC per package taking into account the variability between them</i>
--------------	--

Description

Function to generate scatter plot for each gene in a hunter table and show logFC per package taking into account the variability between them

Usage

```
ht2logFCPlot(  
  ht,  
  var_filter = 0.001,  
  title = "Filtered logFC",  
  top = 50,  
  alpha = 0.5  
)
```

Arguments

ht	: hunter table dataframe
var_filter	: variability threshold to show gene into this graph (low variability will be removed)
title	: plot title
top	: plots only the top N items with more variance. NULL will not filter
alpha	: transparency of dots

Value

plot ready to be rendered

Author(s)

Fernando Moreno Jabato <jabato(at)uma(dot)com>

`install_DEgenes_hunter`*Hunter installation Function*

Description

This function allows you to install scripts.

Usage

```
install_DEgenes_hunter(to)
```

Arguments

`to` Path to folder in which copy Hunter scripts

Value

void

Examples

```
install_DEgenes_hunter("./")
```

`load_and_parse_gmt`*Load a GMT format file and return a dataframe in correct format*

Description

Load a GMT format file and return a dataframe in correct format

Usage

```
load_and_parse_gmt(gmt_file)
```

Arguments

`gmt_file` file to be loaded

Value

GMT loaded info

Examples

```
gmt_file <- system.file("extData",  
  "toy_categories_1.gmt", package = "ExpHunterSuite")  
load_and_parse_gmt(gmt_file)
```

load_synth_dataset	<i>Loads Hunter DEG analysis table and simulated TRUE DEG vector. Concat in training/testing ML data frame format and returns it.</i>
--------------------	---

Description

Loads Hunter DEG analysis table and simulated TRUE DEG vector. Concat in training/testing ML data frame format and returns it.

Usage

```
load_synth_dataset(
  folder = NULL,
  hunter_table_file = "hunter_results_table.txt",
  prediction_file_regex = "_predv"
)
```

Arguments

folder	where files are placed
hunter_table_file	DEG analys result table file name
prediction_file_regex	regex to find TRUE DEG file generated

Value

a data frame with DEG result and real predictions column

Examples

```
# Used to load a synth + DE analysis result folder
ht <- load_synth_dataset()
# Will return an empty object, you need a correct folder
```

load_WGCNA_results	<i>Loads stored WGCNA results in DEgenes Hunter expression analysis package folder and returns in compact object</i>
--------------------	--

Description

Loads stored WGCNA results in DEgenes Hunter expression analysis package folder and returns in compact object

Usage

```
load_WGCNA_results(path, main_deg_table)
```

Arguments

path of expression analysis results (Hunter folder)
 main_deg_table dataframe with main DEG analysis results

Value

WGCNA results loaded

```
main_clusters_to_enrichment
```

Perform enrichment analysis on multiple lists of genes

Description

Perform enrichment analysis on multiple lists of genes

Usage

```
main_clusters_to_enrichment(  
  input_file = NULL,  
  gene_attribute_file = NULL,  
  org_db = NULL,  
  gene_keytype = "ENTREZID",  
  temp_file = "enr_tmp.RData",  
  force = FALSE,  
  all_funsys = c("BP", "MF", "CC"),  
  task_size = 1,  
  current_organism_info = NULL,  
  workers = 1,  
  pvalcutoff = 0.1,  
  qvalcutoff = 0.2,  
  all_custom_gmt = NULL,  
  kegg_data_file = NULL,  
  universe = NULL  
)
```

Arguments

input_file input file path. File must contain 1 cluster per line clusterA ..tab.. gene1,gene2
 gene_attribute_file file to map genes to attributes e.g. fold change
 org_db org_db object

gene_keytype	ID used in the input file
temp_file	enrichments will be read from this file if it exists, written to it if it doesnt
force	if true, to run enrichments even if the temp_file exists
all_funsys	list of GOs/dbs to use for enrichment analysis
task_size	number of elements per packages used
current_organism_info	organism info from text table,
workers	cores to be used if parallel features are going to be used
pvalcutoff	adjusted p-value threshold to use for clusterProfiler,
qvalcutoff	q-value threshold to use for clusterProfiler,
all_custom_gmt	list of custom annotation sets
kegg_data_file	if KEGG included in enrich_dbs path needs
universe	background of genes to use as universe

main_degenes_Hunter	<i>Main DEgenes Hunter Function</i>
---------------------	-------------------------------------

Description

This function allows you to perform the DEG analysis with different algorithms.

Usage

```
main_degenes_Hunter(
  raw = NULL,
  pseudocounts = FALSE,
  target = NULL,
  count_var_quantile = 0,
  external_DEA_data = NULL,
  output_files = getwd(),
  reads = 2,
  minlibraries = 2,
  filter_type = "separate",
  p_val_cutoff = 0.05,
  lfc = 1,
  modules = "DELNW",
  minpack_common = 4,
  model_variables = "",
  numerics_as_factors = TRUE,
  string_factors = NULL,
  numeric_factors = NULL,
  WGCNA_memory = 5000,
  WGCNA_norm_method = "DESeq2",
  WGCNA_deepsplit = 2,
```

```

WGCNA_min_genes_cluster = 20,
WGCNA_detectcutHeight = 0.995,
WGCNA_mergecutHeight = 0.25,
WGCNA_all = FALSE,
WGCNA_blockwiseNetworkType = "signed",
WGCNA_blockwiseTOMType = "signed",
WGCNA_minCoreKME = 0.5,
WGCNA_minCoreKMESize = NULL,
WGCNA_minKMEtoStay = 0.2,
WGCNA_corType = "pearson",
multifactorial = "",
library_sizes = NULL
)

```

Arguments

raw	Dataframe with raw counts per sample and feature
pseudocounts	boolean, activate if the input contains pseudocounts
target	set with control and treatment information
count_var_quantile	quantile variance threshold
external_DEA_data	external DEA set
output_files	otput files path
reads	set of reads
minlibraries	minimum of libraries to use a set of data
filter_type	filter type to ba applied. Allowed: "separate" or "global"
p_val_cutoff	p-value threshold
lfc	minimum logFold Change
modules	modules to be executed. Allowed: DESeq2 (D), NOISeq (N), Limma (L), EdgeR (E), WGCNA (W)
minpack_common	minimum of pack that must be significant to tag a gene as significant
model_variables	custom model
numerics_as_factors	transform numeric values to factors. Default: TRUE
string_factors	string factors for WGCNA
numeric_factors	numeric factors for WGCNA
WGCNA_memory	see WGCNA package
WGCNA_norm_method	see WGCNA package
WGCNA_deepsplit	see WGCNA package

WGCNA_min_genes_cluster
 see WGCNA package

WGCNA_detectcutHeight
 see WGCNA package

WGCNA_mergecutHeight
 see WGCNA package

WGCNA_all see WGCNA package

WGCNA_blockwiseNetworkType
 see WGCNA package

WGCNA_blockwiseTOMType
 see WGCNA package

WGCNA_minCoreKME
 see WGCNA package

WGCNA_minCoreKMESize
 see WGCNA package

WGCNA_minKMEtoStay
 see WGCNA package

WGCNA_corType see WGCNA package

multifactorial specify interaction/effect when multifactorial design

library_sizes NULL or a dataframe with sample names and library sizes

Value

expression analysis result object with studies performed

Examples

```
data(toc)
data(target)
deg_h_out <- main_degenes_Hunter(raw=toc, target=target, modules="D")
```

main_functional_hunter

Perform Functional analysis with different enrichment corpora.

Description

Perform Functional analysis with different enrichment corpora.

Usage

```

main_functional_hunter(
  hunter_results,
  model_organism,
  annot_table = NULL,
  organisms_table = get_organism_table(),
  input_gene_id = "ENSEMBL",
  custom = NULL,
  enrich_dbs = c("MF", "Reactome"),
  kegg_data_file = "",
  enrich_methods = "ORA",
  annotation_source = "orgdb",
  pthreshold = 0.1,
  qthreshold = 0.2,
  cores = 1,
  task_size = 1,
  output_files = "results",
  fc_colname = "mean_logFCs",
  universe = NULL,
  clean_parentals = FALSE,
  simplify = FALSE,
  top_categories = 50,
  sim_thr = NULL,
  summary_common_name = "ancestor",
  clusters_flag = FALSE
)

```

Arguments

hunter_results	output of main_degenes_hunter - list with DEG analysis results
model_organism	organisms which genes are being studied
annot_table	(OPTIONAL) annotation table to translate gene IDs
organisms_table	(OPTIONAL) configuration table for given organism. Use see get_organism_table()
input_gene_id	(OPTIONAL) type of gene IDs given. Allowed: ENSEMBL (E) , entrezgene (e), TAIR/Arabidopsis (T), Gene Names (G).
custom	(OPTIONAL) list of dataframes with GMT corpus
enrich_dbs	annotation dbs to be used: BP,MF,CC,Reactome,KEGG • MF and Reactome default
kegg_data_file	if KEGG included in enrich_dbs path needs
enrich_methods	enrichment methods to be used: ORA or GSEA
annotation_source	annotate with orgdbs or Biomart? Currently on orgdbs
pthreshold	pvalue threshold
qthreshold	qvalue threshold

cores	cores to be used if parallel features are going to be used. Default: 1
task_size	number of elements per packages used
output_files	output folder
fc_colname	main logFC colname (into hunter_results dataframe)
universe	whether to use all genes as the background, or expressed only
clean_parentals	clean parental terms in merged enrichment
simplify	simplify the merged enrichments
top_categories	numbers of categories from each cluster to use for merge
sim_thr	value to use when combining similar categories in summary mode
summary_common_name	'significant' to use the most significant term to label each summarized group
clusters_flag	execute clusters enrichments

Value

functional result object with enrichments performed

Examples

```
# Load DE analysis result
# data(degh_output)
# NOT RUN
# func_results <- main_functional_hunter(hunter_results = degh_output,
# model_organism = "Mouse", enrich_dbs = "MF")
#
```

multienricher_gsea	<i>Perform gsea enrichment analysis of a list of genes</i>
--------------------	--

Description

Perform gsea enrichment analysis of a list of genes

Usage

```
multienricher_gsea(
  all_funsys = NULL,
  genes_list,
  organism_info,
  org_db = NULL,
  task_size = 1,
  workers = 1,
  pvalueCutoff = 0.05,
  pAdjustMethod = "BH",
```

```

    kegg_file = NULL,
    custom_sets = NULL,
    ...
)

```

Arguments

all_funsys	vector of funsys to use (e.g. MF, Reatcome)
genes_list	vector of genes or list of vectors, to be enriched
organism_info	from the annotation table: infor on db names etc
org_db	org db file (optional) - normally obtained from organism_info
task_size	for parallelization
workers	for parallelization
pvalueCutoff	p-value finding enriched categories
pAdjustMethod	adjustment method for p-values
kegg_file	must be provided if kegg annotation required
custom_sets	custom set object (processed by load_and_parse_gmt)
...	other arguments passed to the enrichment function

multienricher_ora	<i>Perform ORA enrichment analysis of a list of genes</i>
-------------------	---

Description

Perform ORA enrichment analysis of a list of genes

Usage

```

multienricher_ora(
  all_funsys = NULL,
  genes_list,
  universe = NULL,
  organism_info,
  org_db = NULL,
  task_size = 1,
  workers = 1,
  pvalueCutoff = 0.05,
  qvalueCutoff = 0.2,
  pAdjustMethod = "BH",
  kegg_file = NULL,
  custom_sets = NULL,
  readable = TRUE,
  return_all = FALSE,
  ...
)

```

Arguments

all_funsys	vector of funsys to use (e.g. MF, Reatcome)
genes_list	vector of genes or list of vectors, to be enriched
universe	background of genes to use as universe
organism_info	from the annotation table: infor on db names etc
org_db	org db file (optional) - normally obtained from organism_info
task_size	for parallelization
workers	for parallelization
pvalueCutoff	p-value finding enriched categories
qvalueCutoff	q-value finding enriched categories
pAdjustMethod	adjustment method for p-values
kegg_file	must be provided if kegg annotation required
custom_sets	custom set object (processed by load_and_parse_gmt)
readable	Whether output should include gene symbols
return_all	Whether to remove list items with no enrichment
...	other arguments passed to the enrichment function

multienricher_topGO	<i>Perform topGO enrichment analysis of a list of genes</i>
---------------------	---

Description

Perform topGO enrichment analysis of a list of genes

Usage

```
multienricher_topGO(
  all_funsys,
  genes_list,
  universe = NULL,
  organism_info,
  gene_id = "entrez",
  algorithm = "classic",
  statistic = "fisher",
  nodeSize = 5,
  task_size = 1,
  workers = 1
)
```

Arguments

all_funsys	vector of funsys to use (e.g. MF, Reatcome)
genes_list	vector of genes or list of vectors, to be enriched
universe	background of genes to use as universe
organism_info	from the annotation table: infor on db names etc
gene_id	what identifier do the genes use - must be entrez for
algorithm	algorithm for detecting enriched functions - see topGO docs
statistic	method for detecting enriched functions
nodeSize	related to the creation of the GOdata object
task_size	for parallelization
workers	for parallelization

parallel_list	<i>Parallelize function execution in list with multiple elements</i>
---------------	--

Description

Parallelize function execution in list with multiple elements

Usage

```
parallel_list(X, FUNC, workers = 2, task_size = 1, ...)
```

Arguments

X	R list
FUNC	Function to execute in each element of the list
workers	Number of process to use in parallel execution
task_size	Number of list elements to be executed on each BiocParallel task
...	Additional arguments to BiocParallel or to FUNC

Value

list

parse_pseudocounts	<i>main_parse_pseudocounts Functions that loads pseudomappers results and complite it into a table of counts</i>
--------------------	--

Description

main_parse_pseudocounts Functions that loads pseudomappers results and complite it into a table of counts

Usage

```
parse_pseudocounts(input_folder, mapper, output_type, annotation_file)
```

Arguments

input_folder	folder with pseudocounts files
mapper	string pseudomapping program
output_type	string indicate if the output must be 'G' genes, 'T' transcripts or both
annotation_file	gtf or gff path, requested for parsin salmon results

Examples

```
## Not run:
parse_pseudocounts(input_folder = "Path_to_folder",
  mapper = "salmon",
  annotation_file = "path_to_gtf")

## End(Not run)
```

ppv	<i>Calculates Positive Predictive Value from a measures dataframe</i>
-----	---

Description

Calculates Positive Predictive Value from a measures dataframe

Usage

```
ppv(df)
```

Arguments

df	dataframe with measures
----	-------------------------

Value

calculated metric

Examples

```
df <- data.frame(TP = 4, FP = 3, TN = 2, FN = 1)
df$Precision <- ppv(df)
```

recall	<i>Calculates Recall from a measures dataframe</i>
--------	--

Description

Calculates Recall from a measures dataframe

Usage

```
recall(df)
```

Arguments

df	dataframe with measures
----	-------------------------

Value

calculated metric

Examples

```
df <- data.frame(TP = 4, FP = 3, TN = 2, FN = 1)
df$Recall <- recall(df)
```

rtable2measures	<i>Calculates emasures from result tables</i>
-----------------	---

Description

Calculates emasures from result tables

Usage

```
rtable2measures(
  htfile,
  realprediction,
  aucfile = NULL,
  experiment = NULL,
  logFC_threshold = 1,
  pval_threshold = 0.01
)
```

Arguments

htfile	hunter table file
realprediction	real predictions file
aucfile	output AUC plot file
experiment	experiment info file to be loaded
logFC_threshold	logFC threshold
pval_threshold	p-value threshold

Value

metrics table

Examples

```
htfile <- system.file("extData",
  "hunter_results_table.txt", package = "ExpHunterSuite")
realprediction <- system.file("extData",
  "ground_truth_DE.txt", package = "ExpHunterSuite")
rtable2measures(htfile, realprediction)
```

scale_data_matrix	<i>Scale a matrix using minimum-maximum method</i>
-------------------	--

Description

Scale a matrix using minimum-maximum method

Usage

```
scale_data_matrix(data_matrix, norm_by_col = FALSE)
```

Arguments

data_matrix	to be scaled
norm_by_col	boolean flag: if true scaling will be performed by columns instead of by rows. Default: FALSE

Value

scaled matrix

scale_range	<i>Funtion used to scale a vector of numeric values from (x_min,x_max) range to (y_min,y_max) range. $f: R \rightarrow R (x_{min}, x_{max}) \rightarrow (y_{min}, y_{max})$ vect - x_min f(vect,y_min,y_max) = $\frac{y_{max} - y_{min}}{x_{max} - x_{min}}$ x (y_max - y_min) + y_min</i>
-------------	--

Description

Funtion used to scale a vector of numeric values from (x_min,x_max) range to (y_min,y_max) range. $f: R \rightarrow R (x_{min}, x_{max}) \rightarrow (y_{min}, y_{max})$ vect - x_min f(vect,y_min,y_max) = $\frac{y_{max} - y_{min}}{x_{max} - x_{min}}$ x (y_max - y_min) + y_min

Usage

scale_range(vect, nmin, nmax)

Arguments

vect	vector to be transformed
nmin	new minimum
nmax	new maximum

Value

transformed vector to new range

spc	<i>Calculates specificity from a measures dataframe</i>
-----	---

Description

Calculates specificity from a measures dataframe

Usage

spc(df)

Arguments

df	dataframe with measures
----	-------------------------

Value

calculated metric

Examples

```
df <- data.frame(TP = 4, FP = 3, TN = 2, FN = 1)
spc(df)
```

STC

Function to Simulate Treatment-Control paired experiments for genes expression count tables. Pipeline inspired on TCC simulateReadCounts function

Description

Function to Simulate Treatment-Control paired experiments for genes expression count tables. Pipeline inspired on TCC simulateReadCounts function

Usage

```
STC(
  Ngene = 1000,
  PDEG = 0.2,
  DEG.foldchange = 2,
  replicates = 3,
  bcount = NULL,
  group = NULL
)
```

Arguments

Ngene	number of genes to be included in the experiment
PDEG	Percentage of Differentially Expressed Genes. If one number is specified, only Up-regulated genes will be simulated. If a vector of two number is given, proportions will be applied for Up and Down regulated simulation respectively
DEG.foldchange	foldchange to be applied. Can be a number or a function which must accept only a vector and return a vector of the same length with the fold-change values. Warning: NOT LOG FOLD CHANGE
replicates	to be generated of each group (Treatment and Control)
bcount	matrix of genes counts used to simulate. If NULL, toc dataset will be loaded
group	columns to be used as group replicates. If NULL, all columns will be used.

Value

a list with COUNT) a simulated gene counts matrix with simulated genes and sample names and TRUEDEG) a vector with real up/down regulated genes

Author(s)

Fernando Moreno Jabato <jabato(at)uma(dot)es>

target	<i>Target data containing sample information for the toc data</i>
--------	---

Description

Target data containing sample information for the toc data

Usage

data(target)

Format

A data.frame of sample information. Rows are samples, columns are different experimental variables/factors

Examples

data(target)

target_generation	<i>Function to generate targets file Load target file if it exists, otherwise use the -C and -T flags. Note target takes precedence over target.</i>
-------------------	--

Description

Function to generate targets file Load target file if it exists, otherwise use the -C and -T flags. Note target takes precedence over target.

Usage

target_generation(from_file = NULL, ctrl_samples = NULL, treat_samples = NULL)

Arguments

from_file	input targets file
ctrl_samples	control samples
treat_samples	target samples

Value

targets info loaded

Examples

```
# To generate target or load from file.
# Example shows generation based on names of treatment/control samples.
# These should match the column names in the table of counts.
case_sample_names <- paste("case", seq(5), sep="_")
control_sample_names <- paste("control", seq(5), sep="_")
target <- target_generation(ctrl_samples=control_sample_names,
                           treat_samples=case_sample_names)
```

toc	<i>Table of counts dummy dataset for DEgeneHunter examples</i>
-----	--

Description

Table of counts dummy dataset for DEgeneHunter examples

Usage

```
data(toc)
```

Format

A truncated data.frame of aligned read counts for genes (rows) and samples (columns)

Examples

```
data(toc)
```

translate_from_table	<i>Translates a given gene ID using a dictionary. Note: one unknown ID can corresponds to many known ids.</i>
----------------------	---

Description

Translates a given gene ID using a dictionary. Note: one unknown ID can corresponds to many known ids.

Usage

```
translate_from_table(ids_to_translate, annot_table)
```

Arguments

```
ids_to_translate    set of IDs to be translated
annot_table         dictionary to translate IDs
```

Value

translated IDs or NA if it's not possible to translate

```
trycatch_pairwise_termsim
```

if pairwise_termsim throws an error, remove dup cat descriptions Only seems to be a problem with Reactome - to study further Further investigation re: number of cats also required

Description

if pairwise_termsim throws an error, remove dup cat descriptions Only seems to be a problem with Reactome - to study further Further investigation re: number of cats also required

Usage

```
trycatch_pairwise_termsim(enr)
```

Arguments

enr enrichment object to be studied

Value

enrichment object after add termsim info

```
write_clusters_to_enrichment
```

Write Main clusters to enrichment output This function allows you to report the Functional analysis.

Description

Write Main clusters to enrichment output This function allows you to report the Functional analysis.

Usage

```
write_clusters_to_enrichment(
  output_path = "results",
  output_file = "results",
  mode = "PR",
  enrichments_ORA = NULL,
  task_size = 1,
  workers = 1,
  template_folder = file.path(find.package("ExpHunterSuite"), "templates"),
```

```

    top_categories = NULL,
    group_results = FALSE,
    n_category = 30,
    sim_thr = 0.7,
    summary_common_name = "ancestor",
    pvalcutoff = 0.1,
    gene_attributes = NULL,
    gene_attribute_name = NULL,
    max_genes = 200,
    simplify = FALSE,
    clean_parentals = FALSE
)

```

Arguments

output_path	output folder
output_file	output file name for heatmaps
mode	type of output to produce - P for plots, R for reports, and S for summarized heatmaps
enrichments_ORA	list of enrich results for all clusters
task_size	number of elements per packages used
workers	(OPTIONAL) cores for parallel features
template_folder	(OPTIONAL) RMD templates folder
top_categories	numbers of categories from each cluster to use for merge
group_results	experimental - whether to group results in the emap plot
n_category	number of categories in the figures (per cluster)
sim_thr	value to use when combining similar categories in summary mode
summary_common_name	'significant' to use the most significant term to label each summarized group 'ancestor' to use the common ancestor of the group"
pvalcutoff	used to select terms for summarizing
gene_attributes	named list of attributes e.g. FCs for emap plot coloured nodes (genes)
gene_attribute_name	name for the legend in the emap plot for the nodes (genes)
max_genes	maximum number of genes to plot in cnet plot
simplify	Activate for process ClusterProfiler results with simplify function
clean_parentals	Activate to reduce significant terms in GO by removing the parentals

Value

void

`write_df_list_as_tables`

Function to write expression package results. Write to disk the results of each diff expression package

Description

Function to write expression package results. Write to disk the results of each diff expression package

Usage

```
write_df_list_as_tables(df_list, prefix, root = getwd())
```

Arguments

<code>df_list</code>	dataframes list
<code>prefix</code>	prefix concatenated to output file name
<code>root</code>	path where write

Value

void

Examples

```
# Typically used with the @all_counts_for_plotting slot of the main output
deg_res <- list(aa=data.frame(a=seq(5),b=seq(5)),
               bb=data.frame(a=seq(6,10),b=seq(6,10)))
write_df_list_as_tables(deg_res, prefix="test", root=tempdir())
```

`write_enrich_files` *Write enrichment files related to functional_hunter results list*

Description

Write enrichment files related to functional_hunter results list

Usage

```
write_enrich_files(func_results, output_path = getwd())
```

Arguments

<code>func_results</code>	functional enrichment results
<code>output_path</code>	output folder path

Value

void

Examples

```
# Use a @functional_hunter or @multienricher result object to  
# create real files  
write_enrich_files(list(), ".")
```

```
write_expression_report
```

Write Main DEgenes Hunter report

Description

This function allows you to report the DEG analysis.

Usage

```
write_expression_report(  
  exp_results,  
  output_files = getwd(),  
  template_folder = NULL,  
  opt = NULL  
)
```

Arguments

exp_results	DEG analysis results object
output_files	output file path
template_folder	RMD templates folder
opt	option list

Value

void

Examples

```
# Load DE analysis results  
degh_output <- list() # data(degh_output)  
write_expression_report(degh_output)
```

write_functional_report

Write Main DEgenes Hunter functional report This function allows you to report the Functional analysis.

Description

Write Main DEgenes Hunter functional report This function allows you to report the Functional analysis.

Usage

```
write_functional_report(
  hunter_results,
  func_results,
  output_files = getwd(),
  fc_colname = "mean_logFCs",
  organisms_table = NULL,
  template_folder = file.path(find.package("ExpHunterSuite"), "templates"),
  cores = 2,
  task_size = 1,
  report = "fc",
  showCategories = 30,
  group_results = FALSE,
  max_genes = 200,
  corr_threshold = 0.8,
  pvalcutoff = 0.05
)
```

Arguments

hunter_results	DEG analysis results
func_results	functional results
output_files	output folder.
fc_colname	(OPTIONAL) main logFC colname (into hunter_results dataframe)
organisms_table	(OPTIONAL) configuration table for given organism. Use see get_organism_table()
template_folder	(OPTIONAL) RMD templates folder
cores	(OPTIONAL) cores for parallel features
task_size	number of elements per packages used
report	string with reports to be written. Allowed: clusters (c)
showCategories	number of categories in the figures (per cluster)

<code>group_results</code>	experimental - whether to group results in the emap plot and functional (f). Default = "fc"
<code>max_genes</code>	maximum number of genes to plot in cnet plot
<code>corr_threshold</code>	minimum module eigengene-trait vector absolute Pearson R value
<code>pvalcutoff</code>	maximum module eigengene-trait vector correlation P value

Value

void

Examples

```
# Load func and DE results
# data(degh_output)
# func_results <- list()
# func_results <- main_functional_hunter(degh_output, "Mouse")
# write_functional_report(degh_output, func_results)
```

Index

- * **datasets**
 - deg_h_output, 3
 - target, 27
 - toc, 28
 - * **design**
 - target_generation, 27
 - * **export**
 - write_enrich_files, 31
 - * **file**
 - load_and_parse_gmt, 11
 - * **input**
 - load_WGCNA_results, 12
 - * **installation**
 - install_DEgenes_hunter, 11
 - * **method**
 - get_organism_table, 8
 - main_degenes_Hunter, 14
 - main_functional_hunter, 16
 - parse_pseudocounts, 22
 - scale_data_matrix, 24
 - * **metrics**
 - acc, 3
 - fmeasure, 6
 - ppv, 22
 - recall, 23
 - rtable2measures, 23
 - spc, 25
 - * **org_db**
 - get_org_db, 7
 - * **output**
 - write_df_list_as_tables, 31
 - * **performance**
 - parallel_list, 21
 - * **synthetic**
 - deg_synth, 4
 - * **translate**
 - translate_from_table, 28
- acc, 3
- deg_h_output, 3
- deg_synth, 4
- download_latest_kegg_db, 5
- extract_from_cm, 5
- fcfunc, 6
- fmeasure, 6
- get_kegg_db_path, 7
- get_org_db, 7
- get_organism_table, 8
- get_stats, 8
- get_stats_from_cm, 9
- get_unexported_function, 9
- ht2logFCPlot, 10
- install_DEgenes_hunter, 11
- load_and_parse_gmt, 11
- load_synth_dataset, 12
- load_WGCNA_results, 12
- main_clusters_to_enrichment, 13
- main_degenes_Hunter, 14
- main_functional_hunter, 16
- multienricher_gsea, 18
- multienricher_oracle, 19
- multienricher_topGO, 20
- parallel_list, 21
- parse_pseudocounts, 22
- ppv, 22
- recall, 23
- rtable2measures, 23
- scale_data_matrix, 24
- scale_range, 25
- spc, 25

STC, [26](#)

target, [27](#)

target_generation, [27](#)

toc, [28](#)

translate_from_table, [28](#)

trycatch_pairwise_termsim, [29](#)

write_clusters_to_enrichment, [29](#)

write_df_list_as_tables, [31](#)

write_enrich_files, [31](#)

write_expression_report, [32](#)

write_functional_report, [33](#)