Package 'dnapath'

October 13, 2022

Type Package

Title Differential Network Analysis using Gene Pathways

Version 0.7.4

Description Integrates pathway information into the differential network analysis of two gene expression datasets as described in Grimes, Potter, and Datta (2019) <doi:10.1038/s41598-019-41918-3>. Provides summary functions to break down the results at the pathway, gene, or individual connection level. The differential networks for each pathway of interest can be plotted, and the visualization will highlight any differentially expressed genes and all of the gene-gene associations that are significantly differentially connected.

Depends R (>= 3.6)

License GPL-2 | GPL-3

LazyData true

Imports Rcpp (>= 1.0.1), corpcor, igraph, ggplot2, SeqNet, Rdpack, dplyr, tibble, stats, methods, curl, graphics, grDevices, gtools, utils, wCorr, parallel

RdMacros Rdpack

LinkingTo Rcpp, RcppArmadillo

Encoding UTF-8

SystemRequirements C++11

RoxygenNote 7.1.2

Suggests testthat (>= 2.1.0), knitr, rmarkdown, markdown, xml2, readr, bc3net, DWLasso, huge, GENIE3, minet, AnnotationDbi, reactome.db, biomaRt

VignetteBuilder knitr

NeedsCompilation yes

Author Tyler Grimes [aut, cre], Somnath Datta [aut]

Maintainer Tyler Grimes <tyler.grimes@unf.edu>

Repository CRAN

Date/Publication 2022-05-09 09:00:05 UTC

R topics documented:

`dnapath2-package` *A short title line describing what the package does*

Description

A more detailed description of what the package does. A length of about one to five lines is recommended.

Details

This section should provide a more detailed overview of how to use the package, including the most important functions.

Author(s)

Your Name, email optional.

Maintainer: Your Name <your@email.com>

References

This optional section can contain literature or other references for background information.

See Also

Optional links to other man pages

Examples

```
## Not run:
   ## Optional simple examples of the most important functions
   ## These can be in \dontrun{} and \donttest{} blocks.

## End(Not run)
```

biomart_hsapiens *Default mapping for entrezgene IDs and HGNC gene symbols*

Description

This dataset is used by default if the connection to biomaRt fails. It is highly recommended to retry the function call that attempted to connect to biomaRt. Using this dataset in general may not produce the correct results.

Usage

```
biomart_hsapiens
```

Format

A data.frame containing a mapping between entrezgene IDs and HGNC gene symbols.

c.dnapath *Combine two 'dnapath' objects.*

Description

This functionality is not implemented and will return an error.

Usage

```
## S3 method for class 'dnapath'
c(...)
```

Arguments

...	'dnapath' objects to be concatenated.

Value

Concatenation is not defined; an error is generated.

c.dnapath_list *Combine two 'dnapath_list' objects.*

Description

This functionality is not implemented and will return an error.

Usage

```
## S3 method for class 'dnapath_list'
c(...)
```

Arguments

...	'dnapath_list' objects to be concatenated.

Value

Concatenation is not defined; an error is generated.

dnapath *Differential Network Analysis Using Gene Pathways*

Description

Integrates pathways into the differential network analysis of gene expression data (Grimes et al. 2019).

Usage

```
dnapath(
  x,
  pathway_list,
  group_labels = NULL,
  network_inference = run_pcor,
  n_perm = 100,
  lp = 2,
  seed = NULL,
  verbose = FALSE,
  mc.cores = 1,
  ...
)
```

Arguments

x	The gene expression data to be analyzed. This can be either (1) a list of two matrices or data frames that contain the gene expression profile from each of two populations (groups) – with rows corresponding to samples and columns to genes – or (2) a single matrix or data frame that contains the expression profiles for both groups. For case (2), the group_labels argument must be specified to identify which rows belong to which group.
pathway_list	A single vector or list of vectors containing gene names to indicate pathway membership. The vectors are used to subset the columns of the matrices in x. A pathway list can be obtained using get_reactome_pathways. If NULL, then the entire expression dataset is analyzed as a single network (this approach is not recommended unless there are only a small number of genes).
group_labels	If x is a single matrix or data frame, group_labels must be specified to label each row. group_labels is a matrix each row corresponding to a in x. This matrix may either (1) have a single column containing the group label for each observation, or (2) individual columns representing each group with values in [0, 1] representing the probability that the patient in that row is in each group. In the latter case, if the rows do not sum to 1, then each entry will be divided by its row sum.
network_inference	A function used to infer the pathway network. It should take in an n by p matrix and return a p by p matrix of association scores. (Built-in options include: run_aracne, run_bc3net, run_c3net, run_clr, run_corr, run_dwlasso, run_genie3, run_glasso, run_mrnet, run_pcor, and run_silencer.) Defaults to run_pcor for partial correlations.
n_perm	The number of random permutations to perform during permutation testing. If n_perm == 1, the permutation tests are not performed. If n_perm is larger than the number of possible permutations, n_perm will be set to this value with a warning message.
lp	The lp value used to compute differential connectivity scores. (Note: If a vector is provided, then the results are returned as a list of dnapath_list objects, one result for each value of lp. This option is available so that network inference methods only need to be run once for each pathway when multple values of lp are being considered. This may be useful when conducting simulation studies).
seed	(Optional) Used to set.seed prior to permutation test for each pathway. This allows results for individual pathways to be easily reproduced.
verbose	Set to TRUE to turn on messages.
mc.cores	Used in mclapply to run the differential network analysis in parallel across pathways. Must be set to 1 if on a Windows machine.
...	Additional arguments are passed into the network inference function.

Value

A 'dnapath_list' or 'dnapath' object containing results for each pathway in pathway_list.

References

Grimes T, Potter SS, Datta S (2019). "Integrating Gene Regulatory Pathways into Differential Network Analysis of Gene Expression Data." *Scientific reports*, **9**(1), 5479.

See Also

filter_pathways, summary.dnapath_list subset.dnapath_list, sort.dnapath_list, plot.dnapath, rename_genes

Examples

```
data(meso)
data(p53_pathways)
set.seed(0)
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways,
                   group_labels = meso$groups, n_perm = 10)
results
summary(results) # Summary over all pathways in the pathway list.
# Remove results for pathways with p-values above 0.2.
top_results <- filter_pathways(results, 0.2)
# Sort the top results by the pathway DC score.
top_results <- sort(top_results, by = "dc_score")
top_results
summary(top_results[[1]])  # Summary of pathway 1.
plot(results[[1]]) # Plot of the differential network for pathway 1.

# Use ... to adjust arguments in the network inference function.
# For example, using run_corr() with method = "spearman":
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways,
                   group_labels = meso$groups, n_perm = 10,
                   network_inference = run_corr,
                   method = "spearman")
results
```

---

d_edgesC *C++ implementation of d_edges*

Description

Calculates differential network score for each edge in a network

Usage

```
d_edgesC(nw1, nw2, lp)
```

Arguments

nw1	The association scores for network 1
nw2	The association scores for network 2
lp	The lp value to use.

Value

A matrix of differential network scores for the edges.

---

d_genesC *C++ implementation of d_genes*

Description

Calculates differential network score for a set of genes

Usage

```
d_genesC(nw1, nw2, lp)
```

Arguments

nw1	The association scores for network 1
nw2	The association scores for network 2
lp	The lp value to use.

Value

A vector of differential network scores for the genes.

---

d_pathwayC *C++ implementation of d_pathway*

Description

Calculates differential network score for an entire pathway.

Usage

```
d_pathwayC(nw1, nw2, lp)
```

Arguments

nw1	The association scores for network 1
nw2	The association scores for network 2
lp	The lp value to use.

Value

The differential network score for the pathway.

`entrez_to_symbol` *Obtain gene symbols for entrezgene IDs*

Description

Uses `biomaRt` (Durinck et al. 2009) to map entrezgene IDs to gene symbols for a given species. Obtains MGI symbols for mouse species and HGNC symbols for other species. (Note: this mapping may not work for all species.) The output of this function can be used in `rename_genes`.

Usage

```
entrez_to_symbol(
  x,
  species,
  symbol_name = NULL,
  dir_save = tempdir(),
  verbose = TRUE
)
```

Arguments

`x`	A vector of entrezgene IDs.
`species`	The species used to obtain the entrezgene IDs. For example: "Homo sapiens", "m musculus", "C. elegans", or "S cerevisiae". "Human" and "mouse" can also be used and will be converted to the correct species name.
`symbol_name`	The type of gene symbol to use. If NULL, then "hgnc_symbol" is used for HGNC symbols, unless `species` is "mmusculus", in which case
`dir_save`	The directory to store annotation reference. Future calls to this function will use the stored annotations. This speeds up the operation and allows for reproducibility in the event that the `biomaRt` database is updated. Set to NULL to disable. By default, it uses a temporary directory to store files during the R session. "mgi_symbol" is used.
`verbose`	Set to FALSE to avoid messages.

Details

If entrezgene IDs are used in a `dnapath_list` or `dnapath` object, or a pathway list, then `get_genes` can be used to extract them and used for the `x` argument here.

Value

A data frame with two columns: the first contains the original entrezgene IDs, and the second contains the corresponding gene symbols. MGI symbols are returned when `species = "Mus musculus"` and HGNC symbols are returned otherwise.

Note

Internet connection is required to connect to biomaRt. If unavailable, the default biomart and default species contained in the package is used, but this may not match the desired species.

References

Durinck S, Spellman PT, Birney E, Huber W (2009). "Mapping Identifiers for the Integration of Genomic Datasets with the R/Bioconductor Package biomaRt." *Nature Protocols*, **4**, 1184–1191.

See Also

symbol_to_entrez, get_genes

Examples

```
data(meso)
# The meso gene expression data contains entrezgene IDs.
# These can be converted to gene symbols.
gene_mat <- entrez_to_symbol(colnames(meso$gene_expression), species = "human")
```

---

filter_pathways *Remove pathways with non-significant DC scores.*

Description

Remove pathways with non-significant DC scores.

Usage

```
filter_pathways(x, alpha_pathway = NULL, monotonized = FALSE)
```

Arguments

x	A 'dnapath_list' object from dnapath.
alpha_pathway	Threshold for pathway p-values to determine significance. If NULL, defaults to 0.05 or the minimum possible threshold (based on the number of permutatiosn that were run).
monotonized	If TRUE, monotonized p-values are used.

Value

A 'dnapath_list' object containing only those pathways with differential connectivity p-values below alpha.

Examples

```
data(meso)
data(p53_pathways)
set.seed(0)
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways,
                   group_labels = meso$groups, n_perm = 10)
results_sig <- filter_pathways(results)
```

`get_genes` *Get the gene names from a differential network analysis*

Description

Get the gene names from a differential network analysis

Usage

```
get_genes(x)
```

Arguments

x	A 'dnapath_list' or 'dnapath' object from dnapath, or a pathway list.

Value

Returns a vector containing all the genes in x.

See Also

rename_genes, entrez_to_symbol, symbol_to_entrez

Examples

```
data(meso)
data(p53_pathways)
set.seed(0)
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways,
                   group_labels = meso$groups, n_perm = 10)
genes <- get_genes(results)
```

get_min_alpha *Get the minimum alpha level for the permutation test*

Description

This method is used internally by several methods to determine the minimum significance threshold (alpha value) that can be applied to the permutation p-values obtained in the differential network analysis.

Usage

```
get_min_alpha(x)
```

Arguments

x	A 'dnapath_list' or 'dnapath' object from dnapath.

Value

The minimum alpha level that can be used based on the number of permutations performed in the analysis.

Examples

```
data(meso)
data(p53_pathways)
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways,
                   group_labels = meso$groups, n_perm = 5)
get_min_alpha(results) # 1 / (5 + 1) = 0.167
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways,
                   group_labels = meso$groups, n_perm = 10)
get_min_alpha(results) # 1 / (10 + 1) = 0.091
```

get_networks *Get the two association networks*

Description

Extracts the estimated association network for each group from the differential network analysis results.

Usage

```
get_networks(x)
```

Arguments

x	A 'dnapath' object from dnapath.

Value

A list of two association matrices.

Note

The two matrices can be plotted using the plot_network function from the SeqNet package, as illustrated in the examples below.

Examples

```
data(meso)
data(p53_pathways)
set.seed(0)
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways,
                   group_labels = meso$groups, n_perm = 10)
# Extract the two estimated association networks for the first pathway
nw <- get_networks(results[[1]])
# Plot the networks using the SeqNet::plot_network function.
# Note that the `compare_graph` argument is used so that the same node layout
# is used across all of the plots.
# Plot the two networks (in separate plots)
g <- SeqNet::plot_network(nw[[1]])
SeqNet::plot_network(nw[[1]], compare_graph = g)
# Plot of the differential network for pathway 1.
# Again, the `compare_graph` argument is used to maintain the same layout.
plot(results[[1]], compare_graph = g)
# We see that genes 51230 and 7311 show strong differential connectivity.
# The plot_pair() function can be used to investigate these two genes further.
plot_pair(results[[1]], "51230", "7311")
```

---

get_reactome_pathways *Obtain Reactome pathways*

Description

Connects to reactome.db (Ligtenberg 2019) to obtain a list of pathways for a given species. The pathway list is processed by combining any two pathways that have substantial overlap (default is over 90% overlap). This output if this function can be used for the pathway_list argument in dnapath.

Usage

```
get_reactome_pathways(
  species,
  overlap_limit = 0.9,
  min_size = 10,
  max_size = 50,
  verbose = TRUE
)
```

Arguments

species	A string, for example "Homo sapiens" or "Mus musculus", indicating the species to use.
overlap_limit	(Optional) Any pathways that have an overlap greater than overlap_limit are combined. Set to NULL to disable this option.
min_size	The minimum pathway size. Any Reactome pathways with fewer than min_size genes are removed from the list. Defaults to 10.
max_size	The maximum pathway size. Any Reactome pathways with more than max_size genes are removed from the list. Defaults to 50.
verbose	Set to FALSE to turn off messages.

Value

A named list of vectors. Each vector corresponds to a Reactome pathway and contains the entrezgene IDs of the genes in that pathway.

References

Ligtenberg W (2019). *reactome.db: A Set of Annotation Maps for Reactome*. R package version 1.68.0.

See Also

The genes in the Reactome pathways use entrezgene IDs. These can be converted to gene symbols, if desired, using the entrez_to_symbol and rename_genes functions.

Examples

```
# Obtaining a pathway list for human (Homo sapiens).
# In this example, overlapping pathways are not combined (this is
# specified by setting overlap_limit to NULL).
pathway_list <- get_reactome_pathways("Homo sapiens", overlap_limit = NULL,
                                      min_size = 10, max_size = 20)
```

head.dnapath_list *Return the first part of the dnapath results.*

Description

Return the first part of the dnapath results.

Usage

```
## S3 method for class 'dnapath_list'
head(x, ...)
```

Arguments

x	A 'dnapath_list' object.
...	Additional paramters are passed into summary.dnapath_list.

Value

Returns the first five rows of the summary table of the 'dnapath_list' object.

Examples

```
data(meso)
data(p53_pathways)
set.seed(0)
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways,
                   group_labels = meso$groups, n_perm = 10)
head(results)
```

length.dnapath_list *The number of pathways in a 'dnapath_list' object.*

Description

The number of pathways in a 'dnapath_list' object.

Usage

```
## S3 method for class 'dnapath_list'
length(x)
```

Arguments

x	A 'dnapath_list' object from dnapath.

Value

The number of pathways.

Examples

```
data(meso)
data(p53_pathways)
set.seed(0)
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways,
                   group_labels = meso$groups, n_perm = 10)
length(results)
```

---

meso *Gene expression dataset for two groups*

Description

meso is a list containing gene expression data from Mesothelioma tumors generated by The Cancer Genome Atlas (TCGA) and obtained using the LinkedOmics portal. The first element in the list, named "gene_expression", contains 32 samples (rows) with 150 genes (columns). The second element, named "groups", is a vector of length 32 indicating which group (stage ii or stage iv) each gene expression sample belongs to. See the "Package data" vignette for details.

Usage

```
meso
```

Format

A list containing two items:

$gene_expression A 32 by 150 matrix of gene expression values

$groups A vector of length 32 indicating which group (stageii or stageiv) each of the rows in the gene expression data belong to.

Source

http://www.linkedomics.org/data_download/TCGA-GBMLGG/

names.dnapath *The pathway names in a 'dnapath' object.*

Description

The pathway names in a 'dnapath' object.

Usage

```
## S3 method for class 'dnapath'
names(x)
```

Arguments

x	A 'dnapath' object from dnapath or from subsetting a 'dnapath_list'.

Value

The pathway's name.

Examples

```
data(meso)
data(p53_pathways)
set.seed(0)
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways,
                   group_labels = meso$groups, n_perm = 10)
names(results[[1]])
```

names.dnapath_list *The pathway names in a 'dnapath_list' object.*

Description

The pathway names in a 'dnapath_list' object.

Usage

```
## S3 method for class 'dnapath_list'
names(x)
```

Arguments

x	A 'dnapath_list' object from dnapath.

Value

The pathway names.

Examples

```
data(meso)
data(p53_pathways)
set.seed(0)
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways,
                   group_labels = meso$groups, n_perm = 10)
names(results)
```

---

p53_pathways *Reactome pathway list for Homo sapiens*

Description

This is a pathway list obtained from get_reactome_pathways with species = "human" (used reactome.db version 1.68.0). Only pathways with "p53" in their name are retained (to subset on some cancer-related pathways). The list contains 13 total pathways. See the "Package data" vignette for details.

Usage

```
p53_pathways
```

Format

A list of 13 vectors each containing a set of entregene IDs.

---

plot.dnapath *Plot function for 'dnapath' object.*

Description

Uses the plotting functions for networks from the SeqNet package (Grimes and Datta 2019)

Usage

```
## S3 method for class 'dnapath'
plot(
  x,
  alpha = NULL,
  monotonized = FALSE,
  only_dc = FALSE,
  require_dc_genes = FALSE,
  scale_edges = 1,
  scale_nodes = 1,
  ...
)
```

Arguments

x	A 'dnapath' object from dnapath.
alpha	Threshold for p-values to infer differentially connected edges. If NULL (the default) then no edges are removed from the plot.
monotonized	If TRUE, monotonized (i.e. step-down) p-values from the permutation test will be used.
only_dc	If TRUE, only differentially connected edges will be shown; any edges that are present in both groups are hidden. If FALSE, the edges shared by both groups are shown. If a non-sparse estimator for network edges is used, then the graph may be dense and setting this argument to TRUE will be useful for highlighting the DC edges.
require_dc_genes	If TRUE, the gene-level differential connectivity p-value of the two genes for a given edge are also considered when deciding whether an edge is differentially connected. If neither gene is significantly differentially connected, then the edge between them will not be either.
scale_edges	(Optional) multiplier for edge widths.
scale_nodes	(Optional) multiplier for node radius
...	Additional arguments are passed into the plotting function plot_network.

Value

Plots the differential network and returns the graph object. See plot_network for details.

References

Grimes T, Datta S (2019). *SeqNet: Generate RNA-Seq Data from Gene-Gene Association Networks*. R package version 1.1.0, https://CRAN.R-project.org/package=SeqNet.

Examples

```
data(meso)
data(p53_pathways)
```

```
set.seed(0)
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways,
                   group_labels = meso$groups, n_perm = 10)
# Plot of the differential network for pathway 1.
plot(results[[1]])
# Plot of the differential network for pathway 1; remove any edges from
# the plot that have p-values above 0.1.
plot(results[[1]], alpha = 0.1)
```

`plot_pair` *Plot the expression values of two genes*

Description

Inspired by the `plotCors` function from the DGCA package, this function is used to plot the expression values of two genes contained in the differential network analysis results. This is useful for comparing the marginal relationship between two genes. Note, however, that this visualization is not able to show conditional associations.

Usage

```
plot_pair(
  x,
  gene_A,
  gene_B,
  method = "loess",
  alpha = 0.5,
  se_alpha = 0.1,
  use_facet = FALSE,
  scales = "fixed"
)
```

Arguments

`x`	A 'dnapath' or 'dnapath_list' object from `dnapath`.
`gene_A`	The name of the first gene to plot. Must be one of the names in `get_genes`(x).
`gene_B`	The name of the second gene to plot. Must be one of the names in `get_genes`(x).
`method`	A charater string, either "lm" or "loess" (the default) used by `geom_smooth` to summarize the marginal gene-gene association. For no line, set method = NULL.
`alpha`	Sets the transparancy of the points, used to set alpha in `geom_point`.
`se_alpha`	Sets the transparancy of the confidence band around the association trend line. Set to 0 to remove the band.
`use_facet`	If TRUE, the groups are plotted in separate graphs using the `link[ggplot2]{facet_wrap}` method.
`scales`	Only used if do_facet_wrap is TRUE. See `link[ggplot2]{facet_wrap}` for details.

Value

Plots the differential network and returns the ggplot object. Additional modifications can be applied to this object just like any other ggplot.

References

Grimes T, Datta S (2019). *SeqNet: Generate RNA-Seq Data from Gene-Gene Association Networks*. R package version 1.1.0, https://CRAN.R-project.org/package=SeqNet.

Examples

```
data(meso)
data(p53_pathways)
set.seed(0)
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways,
                   group_labels = meso$groups, n_perm = 10)
# Plot of the marginal association between the first two genes.
genes <- get_genes(results)[1:2]
g <- plot_pair(results, genes[1], genes[2])
# The ggplot object, g, can be further modified.
# Here we move the legend and use a log scale for the expression values
# (the log scale doesn't help with these data but is shown for demonstration).
g <- g +
  ggplot2::theme(legend.position = "bottom") +
  ggplot2::scale_x_log10() +
  ggplot2::scale_y_log10()
g
```

---

print.dnapath *Print function for 'dnapath' object.*

Description

Print function for 'dnapath' object.

Usage

```
## S3 method for class 'dnapath'
print(x, ...)
```

Arguments

x	A 'dnapath' object from dnapath.
...	Additional arguments are ignored.

Value

Prints a summary of the module.

print.dnapath_list *Print function for 'dnapath_list' object.*

Description

Print function for 'dnapath_list' object.

Usage

```
## S3 method for class 'dnapath_list'
print(x, ...)
```

Arguments

x	A 'dnapath_list' object from dnapath.
...	Additional arguments are ignored.

Value

Prints a summary of the module.

rename_genes *Rename genes in the differential network analysis*

Description

Rename genes in the differential network analysis

Usage

```
rename_genes(x, gene_mat = NULL, to = NULL, species = NULL, ...)
```

Arguments

x	A 'dnapath_list' or 'dnapath' object from dnapath, a pathway list, or a vector of gene names.
gene_mat	(Optional) A matrix of key value pairs. The first column should contain current gene names, and the second column the new names. Any genes that are not in this matrix will retain their current names. This can be any user-defined mapping, or the mapping obtained using entrez_to_symbol or symbol_to_entrez.
to	(Optional) Setting to = "symbol" will rename entrezgene IDs to gene symbols; this will automatically call the entrez_to_symbol() function to obtain the mapping for gene_mat. The species argument must also be specified when to is used.

species	(Optional) Must be specified when setting to = "symbol". This argument is passed into entrez_to_symbol.
...	Additional arugments are passed into entrez_to_symbol in the case that to and species are specified. This may be useful to specify the dir_save argument to save the mapping obtained from biomaRt for offline use.

Value

Returns x with all gene names updated according to gene_mat.

Note

Internet connection is required to connect to use entrez_to_symbol or symbol_to_entrez.

See Also

entrez_to_symbol, symbol_to_entrez

Examples

```
data(meso)
data(p53_pathways)
set.seed(0)
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways,
                   group_labels = meso$groups, n_perm = 10)
summary(results[[1]])  # Summary of pathway 1; note that it uses entrezgene IDs.


# Rename the entrezgene IDs into gene symbols.
results_sym <- rename_genes(results, to = "symbol", species = "human")
summary(results_sym[[1]]) # Now the summary uses gene symbols.
```

---

rev.dnapath_list *Reverse the order of pathways in a 'dnapath_list' object.*

Description

Reverse the order of pathways in a 'dnapath_list' object.

Usage

```
## S3 method for class 'dnapath_list'
rev(x, ...)
```

Arguments

x	A 'dnapath_list' object from dnapath.
...	Additional arguments are ignored.

Value

A 'dnapath_list' object containing the pathways in 'x' in reverse order.

Examples

```
data(meso)
data(p53_pathways)
set.seed(0)
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways,
                   group_labels = meso$groups, n_perm = 10)
# Filter out pathways that have p-values above 0.2.
results <- filter_pathways(results, 0.2)
results <- sort(results, by = "dc_score") # Sort by the pathway DC score.
results <- rev(results) # Reverse the ordering.
```

---

run_aracne *Wrapper for ARACNE method*

Description

Conducts co-expression analysis using ARACNE (Margolin et al. 2006). Uses the implementation from the `minet` package (Meyer et al. 2008). Can be used for the `network_inference` argument in `dnapath`.

Usage

```
run_aracne(
  x,
  weights = NULL,
  estimator = "spearman",
  disc = "none",
  nbins = NULL,
  eps = 0,
  ...
)
```

Arguments

`x`	A n by p matrix of gene expression data (n samples and p genes).
`weights`	An optional vector of weights. This is used by `dnapath()` to apply the probabilistic group labels to each observation when estimating the group-specific network.
`estimator`	Argument is passed into `build.mim`.
`disc`	Argument is passed into `build.mim`.
`nbins`	Argument is passed into `build.mim`.
`eps`	Argument is passed into `aracne`.
`...`	Additional arguments are ignored.

Value

A p by p matrix of association scores.

References

Margolin AA, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Dalla Favera R, Califano A (2006). "ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context." In *BMC Bioinformatics*, volume 7(1), S7. BioMed Central.

Meyer PE, Lafitte F, Bontempi G (2008). "**minet**: A R/Bioconductor Package for Inferring Large Transcriptional Networks using Mutual Information." *BMC Bioinformatics*, **9**(1), 461.

See Also

run_bc3net, run_c3net, run_clr, run_corr, run_dwlasso, run_genie3, run_glasso, run_mrnet, run_pcor, and run_silencer

Examples

```
data(meso)
data(p53_pathways)

# To create a short example, we subset on two pathways from the p53 pathway list,
# and will only run 5 permutations for significance testing.
pathway_list <- p53_pathways[c(8, 13)]
n_perm <- 5

# Use this method to perform differential network analysis.
# The parameters in run_aracne() can be adjusted using the ... argument.
# For example, the 'estimator' parameter can be specified as shown here.
results <- dnapath(x = meso$gene_expression,
                   pathway_list = pathway_list,
                   group_labels = meso$groups,
                   n_perm = n_perm,
                   network_inference = run_aracne,
                   estimator = "spearman")
summary(results)

# The group-specific association matrices can be extracted using get_networks().
nw_list <- get_networks(results[[1]]) # Get networks for pathway 1.


# nw_list has length 2 and contains the inferred networks for the two groups.
# The gene names are the Entrezgene IDs from the original expression dataset.
# Renaming the genes in the dnapath results to rename those in the networks.
# NOTE: The temporary directory, tempdir(), is used in this example. In practice,
#       this argument can be removed or changed to an existing directory
results <- rename_genes(results, to = "symbol", species = "human",
                        dir_save = tempdir())
nw_list <- get_networks(results[[1]]) # The genes (columns) will have new names.

# (Optional) Plot the network using SeqNet package (based on igraph plotting).
```

```
# First rename entrezgene IDs into gene symbols.
SeqNet::plot_network(nw_list[[1]])
```

`run_bc3net` *Wrapper for BC3Net method*

Description

Conducts co-expression analysis using BC3Net (Matos Simoes and Emmert-Streib 2012). Uses the implementation from the `bc3net` package (de Matos Simoes and Emmert-Streib 2016). Can be used for the `network_inference` argument in `dnapath`.

Usage

```
run_bc3net(
  x,
  weights = NULL,
  boot = 100,
  estimator = "spearman",
  disc = "equalwidth",
  mtc1 = TRUE,
  adj1 = "bonferroni",
  alpha1 = 0.05,
  mtc2 = TRUE,
  adj2 = "bonferroni",
  alpha2 = 0.05,
  ...
)
```

Arguments

`x`	A n by p matrix of gene expression data (n samples and p genes).
`weights`	An optional vector of weights. This is used by `dnapath()` to apply the probabilistic group labels to each observation when estimating the group-specific network.
`boot`	Argument is passed into `bc3net`.
`estimator`	Argument is passed into `bc3net`.
`disc`	Argument is passed into `bc3net`.
`mtc1`	Argument is passed into `bc3net`.
`adj1`	Argument is passed into `bc3net`.
`alpha1`	Argument is passed into `bc3net`.
`mtc2`	Argument is passed into `bc3net`.
`adj2`	Argument is passed into `bc3net`.
`alpha2`	Argument is passed into `bc3net`.
`...`	Additional arguments are ignored.

Value

A p by p matrix of association scores.

References

Matos Simoes Rd, Emmert-Streib F (2012). "Bagging Statistical Network Inference from Large-Scale Gene Expression Data." *PloS ONE*, **7**(3), e33624.

de Matos Simoes R, Emmert-Streib F (2016). *bc3net: Gene Regulatory Network Inference with Bc3net*. R package version 1.0.4, https://CRAN.R-project.org/package=bc3net.

See Also

run_aracne, run_c3net, run_clr, run_corr, run_dwlasso, run_genie3, run_glasso, run_mrnet, run_pcor, and run_silencer

Examples

```
data(meso)
data(p53_pathways)

# To create a short example, we subset on one pathway from the p53 pathway list,
# and will only run 1 permutation for significance testing.
pathway_list <- p53_pathways[13]
n_perm <- 1

# Use this method to perform differential network analysis.
# The parameters in run_bc3net() can be adjusted using the ... argument.
# For example, the 'estimator' and 'boot' parameter can be specified as shown here.
results <- dnapath(x = meso$gene_expression,
                   pathway_list = pathway_list,
                   group_labels = meso$groups,
                   n_perm = n_perm,
                   network_inference = run_bc3net,
                   boot = 10,
                   estimator = "pearson",
                   mtc1 = FALSE,
                   mtc2 = FALSE)
summary(results)

# The group-specific association matrices can be extracted using get_networks().
nw_list <- get_networks(results) # Get networks for pathway 1.


# nw_list has length 2 and contains the inferred networks for the two groups.
# The gene names are the Entrezgene IDs from the original expression dataset.
# Renaming the genes in the dnapath results to rename those in the networks.
# NOTE: The temporary directory, tempdir(), is used in this example. In practice,
#       this argument can be removed or changed to an existing directory
results <- rename_genes(results, to = "symbol", species = "human",
                        dir_save = tempdir())
nw_list <- get_networks(results) # The genes (columns) will have new names.
```

```
# (Optional) Plot the network using SeqNet package (based on igraph plotting).
# First rename entrezgene IDs into gene symbols.
SeqNet::plot_network(nw_list[[1]])
```

`run_c3net` *Wrapper for C3Net method*

Description

Conducts co-expression analysis using C3Net (Altay and Emmert-Streib 2010). Uses the implementation from the `bc3net` package (de Matos Simoes and Emmert-Streib 2016). Can be used for the `network_inference` argument in `dnapath`.

Usage

```
run_c3net(
  x,
  weights = NULL,
  estimator = "spearman",
  disc = "equalwidth",
  mtc = TRUE,
  adj = "bonferroni",
  alpha = 0.05,
  ...
)
```

Arguments

`x`	A n by p matrix of gene expression data (n samples and p genes).
`weights`	An optional vector of weights. This is used by `dnapath()` to apply the probabilistic group labels to each observation when estimating the group-specific network.
`estimator`	Argument is passed into `c3mtc`.
`disc`	Argument is passed into `c3mtc`.
`mtc`	Argument is passed into `c3mtc`.
`adj`	Argument is passed into `c3mtc`.
`alpha`	Argument is passed into `c3mtc`.
`...`	Additional arguments are ignored.

Value

A p by p matrix of association scores.

References

Altay G, Emmert-Streib F (2010). "Inferring the Conservative Causal Core of Gene Regulatory Networks." *BMC Systems Biology*, **4**(1), 132.

de Matos Simoes R, Emmert-Streib F (2016). *bc3net: Gene Regulatory Network Inference with Bc3net*. R package version 1.0.4, https://CRAN.R-project.org/package=bc3net.

See Also

run_aracne, run_bc3net, run_clr, run_corr, run_dwlasso, run_genie3, run_glasso, run_mrnet, run_pcor, and run_silencer

Examples

```
data(meso)
data(p53_pathways)

# To create a short example, we subset on one pathway from the p53 pathway list,
# and will only run 1 permutation for significance testing.
pathway_list <- p53_pathways[13]
n_perm <- 1

# Use this method to perform differential network analysis.
# The parameters in run_c3net() can be adjusted using the ... argument.
# For example, the 'estimator' parameter can be specified as shown here.
results <- dnapath(x = meso$gene_expression,
                   pathway_list = pathway_list,
                   group_labels = meso$groups,
                   n_perm = n_perm,
                   network_inference = run_c3net,
                   estimator = "pearson",
                   mtc = FALSE)
summary(results)

# The group-specific association matrices can be extracted using get_networks().
nw_list <- get_networks(results) # Get networks for the pathway.


# nw_list has length 2 and contains the inferred networks for the two groups.
# The gene names are the Entrezgene IDs from the original expression dataset.
# Renaming the genes in the dnapath results to rename those in the networks.
# NOTE: The temporary directory, tempdir(), is used in this example. In practice,
#       this argument can be removed or changed to an existing directory
results <- rename_genes(results, to = "symbol", species = "human",
                        dir_save = tempdir())
nw_list <- get_networks(results) # The genes (columns) will have new names.

# (Optional) Plot the network using SeqNet package (based on igraph plotting).
# First rename entrezgene IDs into gene symbols.
SeqNet::plot_network(nw_list[[1]])
```

run_clr *Wrapper for CLR method*

Description

Conducts co-expression analysis using CLR (Faith et al. 2007). Uses the implementation from the `minet` package (Meyer et al. 2008). Can be used for the `network_inference` argument in `dnapath`.

Usage

```
run_clr(x, weights = NULL, estimator = "spearman", ...)
```

Arguments

`x`	A n by p matrix of gene expression data (n samples and p genes).
`weights`	An optional vector of weights. This is used by `dnapath()` to apply the probabilistic group labels to each observation when estimating the group-specific network.
`estimator`	Argument is passed into `build.mim`.
`...`	Additional arguments are ignored.

Value

A p by p matrix of association scores.

References

Faith JJ, Hayete B, Thaden JT, Mogno I, Wierzbowski J, Cottarel G, Kasif S, Collins JJ, Gardner TS (2007). "Large-Scale Mapping and Validation of Escherichia Coli Transcriptional Regulation from a Compendium of Expression Profiles." *PLoS Biology*, **5**(1), e8.

Meyer PE, Lafitte F, Bontempi G (2008). "**minet**: A R/Bioconductor Package for Inferring Large Transcriptional Networks using Mutual Information." *BMC Bioinformatics*, **9**(1), 461.

See Also

`run_aracne`, `run_bc3net`, `run_c3net`, `run_corr`, `run_dwlasso`, `run_genie3`, `run_glasso`, `run_mrnet`, `run_pcor`, and `run_silencer`

Examples

```
data(meso)
data(p53_pathways)

# To create a short example, we subset on two pathways from the p53 pathway list,
# and will only run 5 permutations for significance testing.
pathway_list <- p53_pathways[c(8, 13)]
```

```
n_perm <- 5

# Use this method to perform differential network analysis.
# The parameters in run_clr() can be adjusted using the ... argument.
# For example, the 'estimator' paramter can be specified as shown here.
results <- dnapath(x = meso$gene_expression,
                   pathway_list = pathway_list,
                   group_labels = meso$groups,
                   n_perm = n_perm,
                   network_inference = run_clr,
                   estimator = "spearman")
summary(results)

# The group-specific association matrices can be extracted using get_networks().
nw_list <- get_networks(results[[1]]) # Get networks for pathway 1.


# nw_list has length 2 and contains the inferred networks for the two groups.
# The gene names are the Entrezgene IDs from the original expression dataset.
# Renaming the genes in the dnapath results to rename those in the networks.
# NOTE: The temporary directory, tempdir(), is used in this example. In practice,
#       this argument can be removed or changed to an existing directory
results <- rename_genes(results, to = "symbol", species = "human",
                        dir_save = tempdir())
nw_list <- get_networks(results[[1]]) # The genes (columns) will have new names.

# (Optional) Plot the network using SeqNet package (based on igraph plotting).
# First rename entrezgene IDs into gene symbols.
SeqNet::plot_network(nw_list[[1]])
```

`run_corr` *Wrapper for correlation co-expression*

Description

Conducts co-expression analysis using correlation for association measure. Can be used for the `network_inference` argument in `dnapath`.

Usage

```
run_corr(
  x,
  weights = NULL,
  threshold = NULL,
  method = c("pearson", "spearman"),
  ...
)
```

Arguments

x	A n by p matrix of gene expression data (n samples and p genes).
weights	An optional vector of weights. This is used by dnapath() to apply the probabilistic group labels to each observation when estimating the group-specific network.
threshold	Cutoff for significant associations. If NULL, all correlations are returned. Otherwise, correlations of magnitude at or below this threshold are set to zero.
method	Argument is passed into cor. Should be one of "pearson" or "spearman".
...	Additional arguments are ignored.

Value

A p by p matrix of association scores.

See Also

run_aracne, run_bc3net, run_c3net, run_clr, run_dwlasso, run_genie3, run_glasso, run_mrnet, run_pcor, and run_silencer

Examples

```
data(meso)
data(p53_pathways)

# To create a short example, we subset on two pathways from the p53 pathway list,
# and will only run 5 permutations for significance testing.
pathway_list <- p53_pathways[c(8, 13)]
n_perm <- 5

# Use this method to perform differential network analysis.
# The parameters in run_corr() can be adjusted using the ... argument.
# For example, the 'method' parameter can be specified as shown here.
results <- dnapath(x = meso$gene_expression,
                   pathway_list = pathway_list,
                   group_labels = meso$groups,
                   n_perm = n_perm,
                   network_inference = run_corr,
                   method = "spearman")
summary(results)

# The group-specific association matrices can be extracted using get_networks().
nw_list <- get_networks(results[[1]]) # Get networks for pathway 1.


# nw_list has length 2 and contains the inferred networks for the two groups.
# The gene names are the Entrezgene IDs from the original expression dataset.
# Renaming the genes in the dnapath results to rename those in the networks.
# NOTE: The temporary directory, tempdir(), is used in this example. In practice,
#       this argument can be removed or changed to an existing directory
results <- rename_genes(results, to = "symbol", species = "human",
```

```
                        dir_save = tempdir())
nw_list <- get_networks(results[[1]]) # The genes (columns) will have new names.

# (Optional) Plot the network using SeqNet package (based on igraph plotting).
# First rename entrezgene IDs into gene symbols.
SeqNet::plot_network(nw_list[[1]])
```

`run_dwlasso` *Wrapper for degree-weighted lasso method*

Description

Conducts co-expression analysis using DWLasso (Sulaimanov et al. 2018). Uses the implementation from the `DWLasso` package (Sulaimanov et al. 2017). Can be used for the `network_inference` argument in `dnapath`.

Usage

```
run_dwlasso(x, weights = NULL, lambda1 = 0.4, lambda2 = 2, ...)
```

Arguments

`x`	A n by p matrix of gene expression data (n samples and p genes).
`weights`	An optional vector of weights. This is used by `dnapath()` to apply the probabilistic group labels to each observation when estimating the group-specific network.
`lambda1`	A penalty parameter that controls degree sparsity of the inferred network. See `DWLasso` for details.
`lambda2`	A penalty parameter that controls overall sparsity of the inferred network. See `DWLasso` for details.
`...`	Additional arguments are ignored.

Value

A p by p matrix of association scores.

References

Sulaimanov N, Kumar S, Burdet F, Ibberson M, Pagni M, Koeppl H (2018). "Inferring Gene Expression Networks with Hubs using a Degree Weighted Lasso Approach." *Bioinformatics*, **35**(6), 987–994.

Sulaimanov N, Kumar S, Koeppl H (2017). *DWLasso: Degree Weighted Lasso*. R package version 1.1, https://CRAN.R-project.org/package=DWLasso.

See Also

run_aracne, run_bc3net, run_c3net, run_clr, run_corr, run_genie3, run_glasso, run_mrnet, run_pcor, and run_silencer

Examples

```
data(meso)
data(p53_pathways)

# To create a short example, we subset on two pathways from the p53 pathway list,
# and will only run 1 permutation for significance testing.
pathway_list <- p53_pathways[c(8, 13)]
n_perm <- 1

# Use this method to perform differential network analysis.
# The parameters in run_dwlasso() can be adjusted using the ... argument.
# For example, the 'lambda1' parameter can be specified as shown here.
results <- dnapath(x = meso$gene_expression,
                   pathway_list = pathway_list,
                   group_labels = meso$groups,
                   n_perm = n_perm,
                   network_inference = run_dwlasso,
                   lambda1 = 0.5)
summary(results)

# The group-specific association matrices can be extracted using get_networks().
nw_list <- get_networks(results[[1]]) # Get networks for pathway 1.


# nw_list has length 2 and contains the inferred networks for the two groups.
# The gene names are the Entrezgene IDs from the original expression dataset.
# Renaming the genes in the dnapath results to rename those in the networks.
# NOTE: The temporary directory, tempdir(), is used in this example. In practice,
#       this argument can be removed or changed to an existing directory
results <- rename_genes(results, to = "symbol", species = "human",
                        dir_save = tempdir())
nw_list <- get_networks(results[[1]]) # The genes (columns) will have new names.

# (Optional) Plot the network using SeqNet package (based on igraph plotting).
# First rename entrezgene IDs into gene symbols.
SeqNet::plot_network(nw_list[[1]])
```

---

run_genie3 *Wrapper for GENIE3 method*

Description

Conducts co-expression analysis using GENIE3 (Huynh-Thu et al. 2010). Uses the implementation from the GENIE3 package. Can be used for the network_inference argument in dnapath.

Usage

```
run_genie3(x, nTrees = 200, weights = NULL, ...)
```

Arguments

x	A n by p matrix of gene expression data (n samples and p genes).
nTrees	Argument is passed into GENIE3.
weights	An optional vector of weights. This is used by dnapath() to apply the probabilistic group labels to each observation when estimating the group-specific network.
...	Additional arguments are ignored.

Value

A p by p matrix of association scores.

References

Huynh-Thu VA, Irrthum A, Wehenkel L, Geurts P (2010). "Inferring Regulatory Networks from Expression Data using Tree-Based Methods." *PloS ONE*, **5**(9), e12776.

See Also

run_aracne, run_bc3net, run_c3net, run_clr, run_corr, run_dwlasso, run_glasso, run_mrnet, run_pcor, and run_silencer

Examples

```
if(!requireNamespace("GENIE3", quietly = TRUE)) {
data(meso)
data(p53_pathways)

# To create a short example, we subset on two pathways from the p53 pathway list,
# and will only run 5 permutations for significance testing.
pathway_list <- p53_pathways[c(8, 13)]
n_perm <- 5

# Use this method to perform differential network analysis.
# The parameters in run_genie3() can be adjusted using the ... argument.
# For example, the 'nTrees' parameter can be specified as shown here.
results <- dnapath(x = meso$gene_expression,
                   pathway_list = pathway_list,
                   group_labels = meso$groups,
                   n_perm = n_perm,
                   network_inference = run_genie3,
                   nTrees = 100)
summary(results)

# The group-specific association matrices can be extracted using get_networks().
nw_list <- get_networks(results[[1]]) # Get networks for pathway 1.
```

```
# nw_list has length 2 and contains the inferred networks for the two groups.
# The gene names are the Entrezgene IDs from the original expression dataset.
# Renaming the genes in the dnapath results to rename those in the networks.
# NOTE: The temporary directory, tempdir(), is used in this example. In practice,
#       this argument can be removed or changed to an existing directory
results <- rename_genes(results, to = "symbol", species = "human",
                        dir_save = tempdir())
nw_list <- get_networks(results[[1]]) # The genes (columns) will have new names.

# (Optional) Plot the network using SeqNet package (based on igraph plotting).
# First rename entrezgene IDs into gene symbols.
SeqNet::plot_network(nw_list[[1]])
}
```

`run_glasso` *Wrapper for glasso method*

Description

Conducts co-expression analysis using glasso (Friedman et al. 2018). Uses the implementation from the huge package (Jiang et al. 2019). Can be used for the `network_inference` argument in `dnapath`.

Usage

```
run_glasso(
  x,
  method = c("glasso", "mb", "ct"),
  criterion = c("ric", "stars"),
  verbose = FALSE,
  weights = NULL,
  ...
)
```

Arguments

`x`	A n by p matrix of gene expression data (n samples and p genes).
`method`	Argument is passed into `huge`.
`criterion`	Argument is passed into `huge.select`.
`verbose`	Argument is passed into `huge` and `huge.select`
`weights`	An optional vector of weights. This is used by `dnapath()` to apply the probabilistic group labels to each observation when estimating the group-specific network.
`...`	Additional arguments are ignored.

Value

A p by p matrix of association scores.

References

Friedman J, Hastie T, Tibshirani R (2018). **glasso***: Graphical Lasso: Estimation of Gaussian Graphical Models*. R package version 1.10.

Jiang H, Fei X, Liu H, Roeder K, Lafferty J, Wasserman L, Li X, Zhao T (2019). *huge: High-Dimensional Undirected Graph Estimation*. R package version 1.3.3, https://CRAN.R-project.org/package=huge.

See Also

run_aracne, run_bc3net, run_c3net, run_clr, run_corr, run_dwlasso, run_genie3, run_mrnet, run_pcor, and run_silencer

Examples

```
data(meso)
data(p53_pathways)

# To create a short example, we subset on one pathway from the p53 pathway list,
# and will only run 1 permutation for significance testing.
pathway_list <- p53_pathways[13]
n_perm <- 1

# Use this method to perform differential network analysis.
# The parameters in run_glasso() can be adjusted using the ... argument.
# For example, the 'criterion' parameter can be specified as shown here.
results <- dnapath(x = meso$gene_expression,
                   pathway_list = pathway_list,
                   group_labels = meso$groups,
                   n_perm = n_perm,
                   network_inference = run_glasso,
                   criterion = "ric")
summary(results)

# The group-specific association matrices can be extracted using get_networks().
nw_list <- get_networks(results) # Get networks for pathway 1.


# nw_list has length 2 and contains the inferred networks for the two groups.
# The gene names are the Entrezgene IDs from the original expression dataset.
# Renaming the genes in the dnapath results to rename those in the networks.
# NOTE: The temporary directory, tempdir(), is used in this example. In practice,
#       this argument can be removed or changed to an existing directory
results <- rename_genes(results, to = "symbol", species = "human",
                        dir_save = tempdir())
nw_list <- get_networks(results) # The genes (columns) will have new names.

# (Optional) Plot the network using SeqNet package (based on igraph plotting).
```

```
# First rename entrezgene IDs into gene symbols.
SeqNet::plot_network(nw_list[[1]])
```

`run_mrnet` *Wrapper for MRNET method*

Description

Conducts co-expression analysis using MRNET (Meyer et al. 2007). Uses the implementation from the `minet` package (Meyer et al. 2008). Can be used for the `network_inference` argument in `dnapath`.

Usage

```
run_mrnet(x, estimator = "spearman", weights = NULL, ...)
```

Arguments

`x`	A n by p matrix of gene expression data (n samples and p genes).
`estimator`	Argument is passed into `build.mim`.
`weights`	An optional vector of weights. This is used by `dnapath()` to apply the probabilistic group labels to each observation when estimating the group-specific network.
`...`	Additional arguments are ignored.

Value

A p by p matrix of association scores.

References

Meyer PE, Kontos K, Lafitte F, Bontempi G (2007). "Information-Theoretic Inference of Large Transcriptional Regulatory Networks." *EURASIP Journal on Bioinformatics and Systems Biology*, **2007**, 8–8.

Meyer PE, Lafitte F, Bontempi G (2008). "**minet**: A R/Bioconductor Package for Inferring Large Transcriptional Networks using Mutual Information." *BMC Bioinformatics*, **9**(1), 461.

See Also

`run_aracne`, `run_bc3net`, `run_c3net`, `run_clr`, `run_corr`, `run_dwlasso`, `run_genie3`, `run_glasso`, `run_pcor`, and `run_silencer`

Examples

```
data(meso)
data(p53_pathways)

# To create a short example, we subset on two pathways from the p53 pathway list,
# and will only run 3 permutations for significance testing.
pathway_list <- p53_pathways[c(8, 13)]
n_perm <- 3

# Use this method to perform differential network analysis.
# The parameters in run_mrnet() can be adjusted using the ... argument.
# For example, the 'estimator' parameter can be specified as shown here.
results <- dnapath(x = meso$gene_expression,
                   pathway_list = pathway_list,
                   group_labels = meso$groups,
                   n_perm = n_perm,
                   network_inference = run_mrnet,
                   estimator = "spearman")
summary(results)

# The group-specific association matrices can be extracted using get_networks().
nw_list <- get_networks(results[[1]]) # Get networks for pathway 1.


# nw_list has length 2 and contains the inferred networks for the two groups.
# The gene names are the Entrezgene IDs from the original expression dataset.
# Renaming the genes in the dnapath results to rename those in the networks.
# NOTE: The temporary directory, tempdir(), is used in this example. In practice,
#       this argument can be removed or changed to an existing directory
results <- rename_genes(results, to = "symbol", species = "human",
                        dir_save = tempdir())
nw_list <- get_networks(results[[1]]) # The genes (columns) will have new names.

# (Optional) Plot the network using SeqNet package (based on igraph plotting).
# First rename entrezgene IDs into gene symbols.
SeqNet::plot_network(nw_list[[1]])
```

`run_pcor` *Wrapper for partial correlations from corpcor*

Description

Conducts co-expression analysis using full partial correlations; these are computed using the shrinkage approach for covariance estimation (Schäfer and Strimmer 2005) from the `corpcor` package (Schafer et al. 2017). Can be used for the `network_inference` argument in `dnapath`.

Usage

```
run_pcor(x, weights = NULL, ranks = FALSE, verbose = FALSE, ...)
```

Arguments

x	A n by p matrix of gene expression data (n samples and p genes).
weights	An optional vector of weights. This is used by dnapath() to apply the probabilistic group labels to each observation when estimating the group-specific network.
ranks	If TRUE, the gene expression values will be converted to ranks (across samples) prior to covariance estimation.
verbose	Argument is passed into pcor.shrink.
...	Additional arguments are ignored.

Value

A p by p matrix of association scores.

References

Schäfer J, Strimmer K (2005). "A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics." *Statistical Applications in Genetics and Molecular Biology*, **4**(1), Article 32.

Schafer J, Opgen-Rhein R, Zuber V, Ahdesmaki M, Silva APD, Strimmer. K (2017). *corpcor: Efficient Estimation of Covariance and (Partial) Correlation*. R package version 1.6.9, https://CRAN.R-project.org/package=corpcor.

See Also

run_aracne, run_bc3net, run_c3net, run_clr, run_corr, run_dwlasso, run_genie3, run_glasso, run_mrnet, and run_silencer

Examples

```
data(meso)
data(p53_pathways)

# To create a short example, we subset on two pathways from the p53 pathway list,
# and will only run 3 permutations for significance testing.
pathway_list <- p53_pathways[c(8, 13)]
n_perm <- 3

# Use this method to perform differential network analysis.
results <- dnapath(x = meso$gene_expression,
                   pathway_list = pathway_list,
                   group_labels = meso$groups,
                   n_perm = n_perm,
                   network_inference = run_pcor)
summary(results)

# The group-specific association matrices can be extracted using get_networks().
nw_list <- get_networks(results[[1]]) # Get networks for pathway 1.
```

```
# nw_list has length 2 and contains the inferred networks for the two groups.
# The gene names are the Entrezgene IDs from the original expression dataset.
# Renaming the genes in the dnapath results to rename those in the networks.
# NOTE: The temporary directory, tempdir(), is used in this example. In practice,
#       this argument can be removed or changed to an existing directory
results <- rename_genes(results, to = "symbol", species = "human",
                        dir_save = tempdir())
nw_list <- get_networks(results[[1]]) # The genes (columns) will have new names.

# (Optional) Plot the network using SeqNet package (based on igraph plotting).
# First rename entrezgene IDs into gene symbols.
SeqNet::plot_network(nw_list[[1]])
```

`run_pcor_fdr` *Wrapper for partial correlations with Empirical Bayes FDR correction*

Description

Conducts co-expression analysis using full partial correlations; these are computed using the shrinkage approach for covariance estimation (Schäfer and Strimmer 2005) from the `corpcor` package (Schafer et al. 2017). Can be used for the `network_inference` argument in `dnapath`. This method will use Empirical Bayes FDR to set some estimates to zero.

Usage

```
run_pcor_fdr(
  x,
  weights = NULL,
  ranks = TRUE,
  thrsh = 1.5,
  verbose = FALSE,
  ...
)
```

Arguments

`x`	A n by p matrix of gene expression data (n samples and p genes).
`weights`	An optional vector of weights. This is used by `dnapath()` to apply the probabilistic group labels to each observation when estimating the group-specific network.
`ranks`	If TRUE, the gene expression values will be converted to ranks (across samples) prior to covariance estimation.
`thrsh`	A positive value (defaults to 1.5). This is used as the cutoff for the likelihood ratio of the estimate local FDR.
`verbose`	Argument is passed into `pcor.shrink`.
`...`	Additional arguments are ignored.

Value

A p by p matrix of association scores.

References

Schäfer J, Strimmer K (2005). "A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics." *Statistical Applications in Genetics and Molecular Biology*, **4**(1), Article 32.

Schafer J, Opgen-Rhein R, Zuber V, Ahdesmaki M, Silva APD, Strimmer. K (2017). *corpcor: Efficient Estimation of Covariance and (Partial) Correlation*. R package version 1.6.9, https://CRAN.R-project.org/package=corpcor.

See Also

run_aracne, run_bc3net, run_c3net, run_clr, run_corr, run_dwlasso, run_genie3, run_glasso, run_mrnet, and run_silencer

Examples

```
data(meso)
data(p53_pathways)

# To create a short example, we subset on two pathways from the p53 pathway list,
# and will only run 3 permutations for significance testing.
pathway_list <- p53_pathways[c(8, 13)]
n_perm <- 3

# Use this method to perform differential network analysis.
results <- dnapath(x = meso$gene_expression,
                   pathway_list = pathway_list,
                   group_labels = meso$groups,
                   n_perm = n_perm,
                   network_inference = run_pcor)
summary(results)

# The group-specific association matrices can be extracted using get_networks().
nw_list <- get_networks(results[[1]]) # Get networks for pathway 1.


# nw_list has length 2 and contains the inferred networks for the two groups.
# The gene names are the Entrezgene IDs from the original expression dataset.
# Renaming the genes in the dnapath results to rename those in the networks.
# NOTE: The temporary directory, tempdir(), is used in this example. In practice,
#       this argument can be removed or changed to an existing directory
results <- rename_genes(results, to = "symbol", species = "human",
                        dir_save = tempdir())
nw_list <- get_networks(results[[1]]) # The genes (columns) will have new names.

# (Optional) Plot the network using SeqNet package (based on igraph plotting).
# First rename entrezgene IDs into gene symbols.
SeqNet::plot_network(nw_list[[1]])
```

`run_silencer` *Wrapper for silencer method*

Description

Conducts co-expression analysis using the matrix silencer method (Barzel and Barabási 2013). Can be used for the `network_inference` argument in `dnapath`.

Usage

```
run_silencer(x, weights = NULL, method = "spearman", verbose = FALSE, ...)
```

Arguments

`x`	A n by p matrix of gene expression data (n samples and p genes).
`weights`	An optional vector of weights. This is used by `dnapath()` to apply the probabilistic group labels to each observation when estimating the group-specific network.
`method`	Argument is passed into `cor`.
`verbose`	If TRUE, updates are printed during the estimation process.
`...`	Additional arguments are ignored.

Value

A p by p matrix of association scores.

References

Barzel B, Barabási A (2013). "Network Link Prediction by Global Silencing of Indirect Correlations." *Nature Biotechnology*, **31**(8), 720.

See Also

`run_aracne`, `run_bc3net`, `run_c3net`, `run_clr`, `run_corr`, `run_dwlasso`, `run_genie3`, `run_glasso`, `run_mrnet`, and `run_pcor`

Examples

```
data(meso)
data(p53_pathways)

# To create a short example, we subset on two pathways from the p53 pathway list,
# and will only run 1 permutation for significance testing.
pathway_list <- p53_pathways[13]
n_perm <- 1
```

```
# Use this method to perform differential network analysis.
# The parameters in run_silencer() can be adjusted using the ... argument.
# For example, the 'method' parameter can be specified as shown here.
results <- dnapath(x = meso$gene_expression,
                   pathway_list = pathway_list,
                   group_labels = meso$groups,
                   n_perm = n_perm,
                   network_inference = run_silencer,
                   method = "spearman")
summary(results)

# The group-specific association matrices can be extracted using get_networks().
nw_list <- get_networks(results) # Get networks for the pathway


# nw_list has length 2 and contains the inferred networks for the two groups.
# The gene names are the Entrezgene IDs from the original expression dataset.
# Renaming the genes in the dnapath results to rename those in the networks.
# NOTE: The temporary directory, tempdir(), is used in this example. In practice,
#       this argument can be removed or changed to an existing directory
results <- rename_genes(results, to = "symbol", species = "human",
                        dir_save = tempdir())
nw_list <- get_networks(results) # The genes (columns) will have new names.

# (Optional) Plot the network using SeqNet package (based on igraph plotting).
# First rename entrezgene IDs into gene symbols.
SeqNet::plot_network(nw_list[[1]])
```

`sort.dnapath_list` *Sort function for 'dnapath_list' object.*

Description

Sort function for 'dnapath_list' object.

Usage

```
## S3 method for class 'dnapath_list'
sort(x, decreasing = TRUE, by = "dc_score", ...)
```

Arguments

`x`	A 'dnapath_list' object from `dnapath`.
`decreasing`	Logical. If TRUE (the default), results are sorted in decreasing order.
`by`	The variable to sort the results by. Must be one of: "mean_expr", the mean expression of each pathway across both groups; "mean_expr1" or "mean_expr2", the mean expression of each pathway in group 1 or 2, respectively; "dc_score",

the differential connectivity score of the pathway; "p_value", the p-value of the dc score; "n_genes", the number of genes in each pathway; "pathway", the pathway names; or "n_dc" the number of significantly differentially conncted genes in each pathway.

`...` Additional arguments are ignored.

Value

The differential network analysis results ordered by DC pathway score.

Examples

```
data(meso)
data(p53_pathways)
set.seed(0)
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways,
                   group_labels = meso$groups, n_perm = 10)
# Filter out pathways that have p-values above 0.2.
results_sig <- filter_pathways(results, 0.2)
sort(results_sig, by = "dc_score") # Sort by the pathway DC score.
sort(results_sig, by = "n_genes") # Sort by the pathway size.
sort(results_sig, by = "mean_expr") # Sort by the mean expression.
```

---

subset.dnapath_list *Subset function for 'dnapath_list' object.*

Description

Subset function for 'dnapath_list' object.

Usage

```
## S3 method for class 'dnapath_list'
subset(x, pathways = NULL, genes = NULL, ...)
```

Arguments

`x` A 'dnapath_list' object from `dnapath`.

`pathways` A set of pathways to index on. This can be (1) a vector of character strings, corresponding to pathway names or regular expressions used to find pathways, (2) a vector of indices to select pathways, (3) a vector of negative indices indicating pathways to remove, or (4) a logical (boolean) vector that is the same length of current number of pathways in `x`.

`genes` A set of gene names to index on; exact matching is used. Only pathways containing these genes are retained.

`...` Additional arguments are ignored.

Value

A subset of the differential network analysis results.

Examples

```
data(meso)
# Obtain a pathway list for this short example:
pathway_list <- get_reactome_pathways("human", overlap_limit = NULL,
                                      min_size = 13, max_size = 19)
# Run the differential network analysis.
results <- dnapath(x = meso$gene_expression, pathway_list = pathway_list,
                   group_labels = meso$groups, n_perm = 5, seed = 0)
summary(results) # Summary over all pathways in the pathway list.

# Subset on pathways that contain "cell cycle" in its name.
cell_cycle_pathways <- subset(results, pathways = "cell cycle")
summary(cell_cycle_pathways)
# Subset on pathways that contain the gene 1026 (Entrezgene ID).
pathways_with_1026 <- subset(results, genes = "1026")
summary(pathways_with_1026)

# Multiple pathways and/or genes can also be specified.
# Specifying both acts as an "OR" operation. For example, the following subset
# will contain pathways containing the words "acetylation" or "methylation"
# OR pathways that contain the genes "1108" or "11200".
results_OR <- subset(results,
                     pathways = c("acetylation", "methylation"),
                     genes = c("1108", "11200"))
summary(results_OR)
# To subset on pathways that have both a specific pathway name AND
# certain genes, call the subset function twice: once specifying the
# `pathways` argument, then pass those results back into subset() with the
# `genes` argument specified. For example:
results_AND <- subset(results,
                      pathways = c("acetylation", "methylation"))
results_AND <- subset(results_AND,
                      genes = c("1108", "11200"))
summary(results_AND)
```

---

summarize_edges *Summarize differential connections for a pathway*

Description

Summarize differential connections for a pathway

Usage

```
summarize_edges(x, alpha = 0.1, monotonized = FALSE, require_dc_genes = FALSE)
```

Arguments

x	A 'dnapath' object from dnapath.
alpha	Threshold for p-values of edge DC scores. Defaults to 0.1 or the minimum possible threshold for the number of permutations performed, whichever is greater.
monotonized	If TRUE, monotonized p-values are used.
require_dc_genes	If TRUE, the gene-level differential connectivity p-value of the two genes for a given edge are also considered when deciding whether an edge is differentially connected. If neither gene is significantly differentially connected, then the edge between them will not be either.

Value

A tibble summarizing the differential connections in the pathway.

See Also

summarize_pathways, summarize_genes

Examples

```
data(meso)
data(p53_pathways)
set.seed(0)
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways,
                   group_labels = meso$groups, n_perm = 10)
summarize_edges(results[[1]])
```

---

summarize_genes *Summarize the differential connectivity of genes over all pathways.*

Description

Summarize the differential connectivity of genes over all pathways.

Usage

```
summarize_genes(x, alpha = 0.1, monotonized = FALSE)
```

Arguments

x	A 'dnapath_list' object from dnapath.
alpha	Threshold for p-values of gene DC scores. Used to determine the number of pathways that each gene is differentially connected in. Defaults to 0.1 or the minimum possible threshold for the number of permutations performed, whichever is greater.
monotonized	If TRUE, monotonized p-values are used.

Value

A tibble summarizing the differential connectivity of genes across all pathways.

See Also

summarize_pathways, summarize_edges

Examples

```
data(meso)
data(p53_pathways)
set.seed(0)
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways,
                   group_labels = meso$groups, n_perm = 10)
summarize_genes(results) # Summary of genes across all pathways.
summarize_genes(results[[1]]) # Summary of genes within the first pathway.
```

---

summarize_pathways *Summarize the differential connectivity of pathways.*

Description

Summarize the differential connectivity of pathways.

Usage

```
summarize_pathways(x, alpha = 1, alpha_gene = 0.1, monotonized = FALSE)
```

Arguments

x	A 'dnapath_list' object from dnapath.
alpha	Threshold for p-values of pathway DC scores. Defaults to 1, which leads to results for all pathways being shown.
alpha_gene	Threshold for p-values of gene DC scores. Used to determine the number of genes that are differentially connected within each pathway. Defaults to 0.1 or the minimum possible threshold for the number of permutations performed, whichever is greater.
monotonized	If TRUE, monotonized p-values are used.

Value

A tibble summarizing the differential connectivity of genes in the pathway.

See Also

summarize_genes, summarize_edges

Examples

```
data(meso)
data(p53_pathways)
set.seed(0)
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways,
                   group_labels = meso$groups, n_perm = 10)
summarize_pathways(results)
```

`summary.dnapath` *Summary function for 'dnapath' object.*

Description

Summary function for 'dnapath' object.

Usage

```
## S3 method for class 'dnapath'
summary(object, by_gene = TRUE, alpha = 1, monotonized = FALSE, ...)
```

Arguments

`object`	A 'dnapath' object from `dnapath`.
`by_gene`	If TRUE, summarizes the differential network analysis by genes; otherwise, summarizes by gene-gene interactions.
`alpha`	Threshold for p-values to determine significance; defaults to 1 and returns all results. If 'by_gene' is FALSE, then 'alpha' is used to filter edges. If 'by_gene' is TRUE, then 'alpha' is used to filter genes.
`monotonized`	If TRUE, monotonized p-values are used.
`...`	Additional arguments are ignored.

Value

Summarizes the differential network analysis result.

See Also

`summarize_genes`, `summarize_edges`

Examples

```
data(meso)
data(p53_pathways)
set.seed(0)
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways,
                   group_labels = meso$groups, n_perm = 10)
summary(results[[1]]) # Summary of the first pathway in the results.
```

summary.dnapath_list *Summary function for 'dnapath_list' object.*

Description

Summary function for 'dnapath_list' object.

Usage

```
## S3 method for class 'dnapath_list'
summary(
  object,
  by_gene = FALSE,
  alpha_pathway = 1,
  alpha_gene = 0.1,
  monotonized = FALSE,
  ...
)
```

Arguments

object	A 'dnapath_list' object from dnapath.
by_gene	If TRUE, summarizes the differential network analysis by genes instead of by pathways.
alpha_pathway	Threshold for p-values of pathway DC scores; used to subset the results. If NULL (or 1), results for all pathways are shown.
alpha_gene	Threshold for p-values of gene DC scores. Used to determine the number of genes that are differentially connected within each pathway. Defaults to 0.1 or the minimum possible threshold for the number of permutations performed, whichever is greater.
monotonized	If TRUE, monotonized p-values are used.
...	Additional arguments are ignored.

Value

Summarizes the differential network analysis results.

See Also

summarize_pathways, summarize_genes

Examples

```
data(meso)
data(p53_pathways)
set.seed(0)
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways,
                   group_labels = meso$groups, n_perm = 10)
summary(results) # Summary across all pathways in the analysis.
```

`symbol_to_entrez` *Obtain entrezgene IDs for gene symbols*

Description

Uses `biomaRt` (Durinck et al. 2009) to map entrezgene IDs to gene symbols for a given species. The output of this function can be used in `rename_genes`.

Usage

```
symbol_to_entrez(
  x,
  species,
  symbol_name = NULL,
  dir_save = tempdir(),
  verbose = TRUE
)
```

Arguments

`x`	A vector of gene symbols.
`species`	The species used to obtain the entrezgene IDs. For example: "Homo sapiens", "m musculus", "C. elegans", or "S cerevisiae". "Human" and "mouse" can also be used and will be converted to the correct species name.
`symbol_name`	The type of gene symbol to use. If NULL, then "hgnc_symbol" is used for HGNC symbols, unless `species` is "mmusculus", in which case "mgi_symbol" is used.
`dir_save`	The directory to store annotation reference. Future calls to this function will use the stored annotations. This speeds up the operation and allows for reproducibility in the event that the `biomaRt` database is updated. Set to NULL to disable. By default, it uses a temporary directory to store files during the R session.
`verbose`	Set to FALSE to avoid messages.

Details

If entrezgene IDs are used in a `dnapath_list` or `dnapath` object, or a pathway list, then `get_genes` can be used to extract them and used for the `x` argument here.

Value

A data frame with two columns: the first contains the original gene symbols, and the second contains a corresponding entrezgene ID. If a gene symbol is not mapped to an entrezgene ID, the entrezgene ID is set to -1.

Note

Internet connection is required to connect to biomaRt. If unavailable, the default biomart and default species contained in the package is used, but this may not match the desired species.

It is assumed that x contains MGI symbols when the biomart species is "Mus musculus" and HGNC symbols otherwise.

References

Durinck S, Spellman PT, Birney E, Huber W (2009). "Mapping Identifiers for the Integration of Genomic Datasets with the R/Bioconductor Package biomaRt." *Nature Protocols*, **4**, 1184–1191.

See Also

entrez_to_symbol, get_genes

Examples

```
# Convert a set of gene symbols to entrezgene IDs.
# Note that not all may have mapping (such as "MSX" in this example).
gene_mat <- symbol_to_entrez(c("SOX2", "SEMA3E", "COL11A1", "UBB", "MSX"),
                             species = "human")
```

---

tail.dnapath_list *Return the last part of the dnapath results.*

Description

Return the last part of the dnapath results.

Usage

```
## S3 method for class 'dnapath_list'
tail(x, ...)
```

Arguments

x	A 'dnapath_list' object.
...	Additional paramters are passed into summary.dnapath_list.

Value

Returns the last five rows of the summary table of the 'dnapath_list' object.

Examples

```
data(meso)
data(p53_pathways)
set.seed(0)
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways,
                   group_labels = meso$groups, n_perm = 10)
tail(results)
```

[.dnapath *Extract results of a single pathway from a 'dnapath' object.*

Description

Extract results of a single pathway from a 'dnapath' object.

Usage

```
## S3 method for class 'dnapath'
x[i, ...]
```

Arguments

x	A 'dnapath' object.
i	The index specifying which pathway to extract.
...	Additional arguments are ignored.

Value

The 'dnapath' object unmodified

Note

In the current implementation, there is nothing to subset on for individual pathway results, so the original object is returned unmodified.

Examples

```
data(meso)
data(p53_pathways)
set.seed(0)
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways[[1]],
                   group_labels = meso$groups, n_perm = 10)
results[1]
```

[.dnapath_list *Extract parts of a 'dnapath_list' object.*

Description

Extract parts of a 'dnapath_list' object.

Usage

```
## S3 method for class 'dnapath_list'
x[i, ...]
```

Arguments

x	A 'dnapath_list' object from dnapath.
i	The indices of pathways to extract.
...	Additional arguments are ignored.

Value

A 'dnapath_list' object containing pathways indexed by 'i'.

[<-.dnapath *Replace parts of a 'dnapath' object.*

Description

This functionality is not implemented and will return an error.

Usage

```
## S3 replacement method for class 'dnapath'
x[...] <- value
```

Arguments

x	A 'dnapath' object from dnapath.
...	Additional arguments are ignored.
value	A 'dnapath' object.

Value

Replacement is not defined; an error is generated.

[<-.dnapath_list *Replace parts of a 'dnapath_list' object.*

Description

This functionality is not implemented and will return an error.

Usage

```
## S3 replacement method for class 'dnapath_list'
x[...] <- value
```

Arguments

x	A 'dnapath_list' object from dnapath.
...	Additional arguments are ignored.
value	A 'dnapath_list' object.

Value

Replacement is not defined; an error is generated.

[[.dnapath *Extract results of a single pathway from a 'dnapath' object.*

Description

Extract results of a single pathway from a 'dnapath' object.

Usage

```
## S3 method for class 'dnapath'
x[[i, ...]]
```

Arguments

x	A 'dnapath' object.
i	The index specifying which pathway to extract.
...	Additional arguments are ignored.

Value

The 'dnapath' object unmodified

Note

In the current implementation, there is nothing to subset on for individual pathway results, so the original object is returned unmodified.

Examples

```
data(meso)
data(p53_pathways)
set.seed(0)
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways[[1]],
                   group_labels = meso$groups, n_perm = 10)
results[[1]]
```

[[.dnapath_list *Extract results of a single pathway from a 'dnapath_list' object.*

Description

Extract results of a single pathway from a 'dnapath_list' object.

Usage

```
## S3 method for class 'dnapath_list'
x[[i, ...]]
```

Arguments

x	A 'dnapath_list' object from dnapath.
i	The index specifying which pathway to extract.
...	Additional arguments are ignored.

Value

A 'dnapath' object containing a single pathway result.

Examples

```
data(meso)
data(p53_pathways)
set.seed(0)
results <- dnapath(x = meso$gene_expression, pathway_list = p53_pathways,
                   group_labels = meso$groups, n_perm = 10)
results[[1]]
```

Index