Package ‘PFP’

March 10, 2023

Type Package

Title Pathway Fingerprint Framework in R

Version 1.7.0

biocViews Software, Pathways, RNASeq

Description An implementation of the pathway fingerprint framework that introduced in paper ``Pathway Fingerprint: a novel pathway knowledge and topology based method for biomarker discovery and characterization".
This method provides a systematic comparisons between a gene set (such as a list of differentially expressed genes) and well-studied ``basic pathway networks" (KEGG pathways), measuring the importance of pathways and genes for the gene set.
The package is helpful for researchers to find the biomarkers and its function.

Depends R (>= 4.0)

Imports graph, igraph, KEGGgraph, clusterProfiler, ggplot2, plyr, tidyr, magrittr, stats, methods, utils

Suggests knitr, testthat, rmarkdown, org.Hs.eg.db

License GPL-2

Encoding UTF-8

LazyData FALSE

VignetteBuilder knitr

NeedsCompilation no

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

URL https://github.com/aib-group/PFP

BugReports https://github.com/aib-group/PFP/issues

git_url https://git.bioconductor.org/packages/PFP

git_branch master

git_last_commit c11fb9c

git_last_commit_date 2022-11-01

Date/Publication 2023-03-10

Author XC Zhang [aut, cre]

Maintainer XC Zhang <`kunghero@163.com`>

R topics documented:

`calc_PFP_score` *Get the pathway fingerprint of a gene_list*

Description

It can evaluate the performance of a gene list in the pathway networks.

Usage

```
calc_PFP_score(
  genes,
  PFPRefnet,
  lambda = 0.5,
  coeff1 = 1,
  coeff2 = 0.1,
  statistic = TRUE,
  bg_genelist = NULL,
  adjust_method = "BH"
)
```

Arguments

`genes,`	a vector of characters
`PFPRefnet,`	A PFPRefnet class
`lambda,`	a numeric, the coefficient for keeping balance between the node_score and edge_score in PFP model
`coeff1,`	a numeric, the weight coefficient for directly connected score in PFP model
`coeff2,`	a numeric, the weight coefficient for indirectly connected score in PFP model
`statistic,`	a logical,whether to do the statistical test
`bg_genelist,`	a vector of characters, background gene set for the statistical test
`adjust_method,`	statistic test method for adjust the p_value. It could be "holm", "hochberg", "hommel", "bonferroni", "BH", "BY","fdr", "none".

Details

The main part of pathway fingerprint. PFP is used to evaluate the performance of a gene_list in some pathway networks by considering the genes' topological location in a pathway. Then we can get every gene's score and the pathway score is caculated by sum all genes' score. All pathways' scores combine the pathway fingerprint.

Value

The score of PFP

Examples

```
data(gene_list_hsa)
data(PFPRefnet_hsa)
PFP <- calc_PFP_score(gene_list_hsa,PFPRefnet_hsa)
```

`data_std` *A matrix of counts A dataset of gene expression profile, a large matrix with 21 rows and 2603 columns.*

Description

A matrix of counts A dataset of gene expression profile, a large matrix with 21 rows and 2603 columns.

Format

Gene list obtained by differential gene analysis

Examples

```
data(data_std)
```

`genes_score-methods` *The score of genes in* PFP *class*

Description

This function extract the detail scores of every gene in the gene_list by specific condition.

Usage

```
genes_score(
  object,
  index = NULL,
  index_type = c("pathway_id", "pathway_name", "slice")
)

## S4 method for signature 'PFP'
genes_score(
  object,
  index = NULL,
  index_type = c("pathway_id", "pathway_name", "slice")
)
```

Arguments

`object,`	PFP class
`index,`	character, indicating the groups to subset.
`index_type,`	"pathway_id","pathway_name","slice"

Value

a named vector of numeric scores

See Also

`PFP-class`

Examples

```
data(PFP_test1)
genes_score <- genes_score(PFP_test1)
```

`gene_list_hsa` *A gene list of human gene_list_hsa is a array of 40 genetic ENTREZID.*

Description

A gene list of human gene_list_hsa is a array of 40 genetic ENTREZID.

Format

Gene list obtained by differential gene analysis

Examples

```
data(gene_list_hsa)
```

`get_asso_net` *merge the edges_coexp and edges_kegg*

Description

This function will remove the co-expressed edges in edges_coexp which also emerge in edges_kegg.

Usage

```
get_asso_net(
  edges_coexp,
  edges_kegg,
  file_dir = NULL,
  if_symbol = TRUE,
  trans_fun = trans_edges_id,
  from_type = "ENTREZID",
  to_type = "SYMBOL",
  gene_info_db = NULL
)
```

Arguments

`edges_coexp,`	a data.frame whose colnames is "source","target", "weight","pathway","edge_type".
`edges_kegg,`	a data.frame whose colnames is "source","target", "weight","pathway","edge_type".
`file_dir,`	a character, the root to save the result of nodes & edges.
`if_symbol,`	a logical,whether to translate the gene id type. Default is TRUE.
`trans_fun,`	a function, when `if_symbol` is *TRUE*,it will use the `trans_fun` function to translate the gene ids. Default is `trans_edges_id`.
`from_type,`	a character,the parameter in `trans_fun`. It is the type of gene ID, "ENSEMBL","GO","SYMBOL" and so on.
`to_type,`	a character,the parameter in `trans_fun`. It is the type of gene ID, "ENSEMBL","GO","SYMBOL" and so on.
`gene_info_db,`	an AnnotationDb-object for gene annotation, such as "org.Hs.eg.db".

Details

This function will remove the co-expressed edges in edges_coexp which also emerge in edges_kegg. It will return a list contains two data.frames. One is the merged data. Another is the nodes information of the edges.

Value

the nodes information of the edges.

Examples

```
data(PFPRefnet_hsa)
data(data_std)
data(PFP_test1)
rank1 <- rank_PFP(object = PFP_test1,total_rank = TRUE)
pathway_select <- refnet_info(rank1)[1,"id"]
gene_test <- pathways_score(rank1)$genes_score[[pathway_select]]$ENTREZID
edges_coexp <- get_exp_cor_edges(gene_test,data_std)
gene_list2 <- unique(c(edges_coexp$source,edges_coexp$target))
edges_kegg <- get_bg_related_kegg(gene_list2,
                                  PFPRefnet=PFPRefnet_hsa,
                                  rm_duplicated = TRUE)
```

`get_bg_related_kegg` *get_bg_related_kegg*

Description

This function will select all genes in all kegg pathways which are directly connected with the genes in `gene_list`

Usage

```
get_bg_related_kegg(gene_list, PFPRefnet, rm_duplicated = FALSE)
```

Arguments

`gene_list,`	a vector of characters, refers to genes ids
`PFPRefnet,`	an object of PFPRefnet class, it contains all kegg pathways.
`rm_duplicated,`	a logical, whether to remove the duplicated kegg edges in different pathways. Defalut is *FALSE*

Details

It will return a data.frame which can be translated a graph or network. In the data.frame, `source` refers to the genes in `gene_list`, `target` refers to the directly connected genes in kegg, `weight` is 0.5, no real means, `pathway`refers to the pathway which the edge emerge and `edge_type` is "kegg".Note, if `rm_duplicated` is *FALSE*, it may return many duplicated edges,which will be complex when plotting a network. If `rm_duplicated` is *TRUE*,it will retain the first pathway which contains the duplicated edge.

Value

the related kegg network.

Examples

```
data(PFPRefnet_hsa)
data(gene_list_hsa)
edges_kegg <- get_bg_related_kegg(gene_list_hsa,
                                  PFPRefnet=PFPRefnet_hsa,
                                  rm_duplicated = TRUE)
```

`get_exp_cor_edges` *get co-expression genes*

Description

compute the correlation coefficient of gene expression data, return the most related genes

Usage

```
get_exp_cor_edges(
  gene_list,
  data_std,
  method = "spearman",
  num = 5,
  cor_threshold = NULL
)
```

Arguments

`gene_list,`	a vector of characters
`data_std,`	a matrix of data, such as gene expression data, whose rownames are gene names or ids and colnames are sample names
`method,`	a chareater, which method to compare the correlation of gene expression data it could be "pearson", "kendall", "spearman", "spearman" is default
`num,`	an integer, the top number of co-expressed genes to choose, 5 is default
`cor_threshold,`	a numeric, the threshold of the correlation coefficient to choose, default is *NULL*

Details

This function computes the correlation coefficient of gene expression data between `gene_list` and `data_std`, it will return a data.frame which can be translated a graph or network. In the data.frame, `source` refers to the genes in `gene_list`, `target` refers to the top coexpressed genes, `weight` refers to the correlated coefficient of genes in `source` and `target`, `pathway` is "uncertain" and `edge_type` is "coexp".Note, when choosing the top co-expressed genes, we will use the `num` param if the `cor_threshold` param is *NULL*. If not, we will choose the `cor_threshold` param.

Value

the coexp of edges.

Examples

```
data(data_std)
data(PFP_test1)
rank1 <- rank_PFP(object = PFP_test1,total_rank = TRUE)
pathway_select <- refnet_info(rank1)[1,"id"]
gene_test <- pathways_score(rank1)$genes_score[[pathway_select]]$ENTREZID
edges_coexp <- get_exp_cor_edges(gene_test,data_std)
```

`get_pathway_info` *get pathway info of a species in KEGG*

Description

This function helps get pathway info of a species in KEGG.

Usage

```
get_pathway_info(spec)
```

Arguments

`spec,`	a character, refers to the species in KEGG. hsa, mmu...

Details

, get pathway info of a species in KEGG. It will return a data.frame.

Value

a data.frame whose colnames contains "index","id","name" and "group"

Examples

```
pathway_info <- get_pathway_info("hsa")
```

`get_PFPRefnet` *get a PFPRefnet for a species*

Description

This function helps update the latest PFPRefnet odject for a species

Usage

```
get_PFPRefnet(spec, file_root = ".", test_mode = FALSE)
```

Arguments

`spec,`	a character, refers to the species in KEGG. hsa, mmu...
`file_root,`	a character, file dir to download the kgml files.
`test_mode,`	please set whether to test this function.

Details

, gupdate the latest PFPRefnet odject for a species in KEGG. It will return a PFPRefnet object.

Value

a PFPRefnet object.

Examples

```
PFPRefnet1 <- get_PFPRefnet("hsa",".",test_mode=TRUE)
```

`group-methods` *group information of* PFPRefnet

Description

This function contains names of basic groups of the networks and group number, as well as the size of each group

Usage

```
group(object)

## S4 method for signature 'PFPRefnet'
group(object)
```

Arguments

`object,`	`PFPRefnet` class

Value

a list contains names of basic groups of the networks and group number, as well as the size of each group

See Also

`PFPRefnet-class`

Examples

```
data(PFPRefnet_hsa)
group <- group(PFPRefnet_hsa)
```

`kegg_download` *download kegg KGML files*

Description

This function will download all kegg KGML files assigned by `spec`.

Usage

```
kegg_download(spec, file_root = ".", test_mode = FALSE)
```

Arguments

`spec,`	a character, refers to the species names in kegg, such as "hsa", "mmu"...
`file_root,`	a character,refers to the root you want to save kegg pathway kgml files in.
`test_mode,`	please set whether to test this function.

Details

Downloading all kegg KGML files assigned by `spec` from https://www.kegg.jp/kegg/xml/, which may take tens of minutes.

Value

the kegg KGML files

Examples

```
kegg_download(spec,file_root=".", test_mode=TRUE)
```

`network-methods` *Basic pathway networks of* PFPRefnet *class*

Description

This function extract the basic networks of PFPRefnet class.

Usage

```
network(object)

## S4 method for signature 'PFPRefnet'
network(object)
```

Arguments

`object,`	`PFPRefnet` class

Value

a graphNEL list of all basic networks

See Also

`PFPRefnet-class`

Examples

```
data(PFPRefnet_hsa)
network <- network(PFPRefnet_hsa)
```

`net_info-methods` *Basic pathway networks information of* PFPRefnet *class*

Description

This function extract the basic networks information of PFPRefnet class.

Usage

```
net_info(object)

## S4 method for signature 'PFPRefnet'
net_info(object)
```

Arguments

`object,`	`PFPRefnet` class

Value

a dataframe contains basic networks' information

See Also

`PFPRefnet-class`

Examples

```
data(PFPRefnet_hsa)
net_info <- net_info(PFPRefnet_hsa)
```

`net_names-methods` *Names of basic networks*

Description

This function extract the network names of PFPRefnet.

Usage

```
net_names(object)

## S4 method for signature 'PFPRefnet'
net_names(object)
```

Arguments

`object,`	`PFPRefnet` class

Value

a vector contains pathway names

See Also

`PFPRefnet-class`

Examples

```
data(PFPRefnet_hsa)
net_names <- net_names(PFPRefnet_hsa)
```

`pathways_score-methods` *Basic pathway networks scores of* PFP *class*

Description

This function can extract the details in pathway fingerprint scores.

Usage

```
pathways_score(object)

## S4 method for signature 'PFP'
pathways_score(object)
```

Arguments

`object,`	`PFP` class

Value

as list, details in pathway fingerprint scores.

See Also

`PFP-class`

Examples

```
data(PFP_test1)
pathways_score <- pathways_score(PFP_test1)
```

`pathway_info` *pathway_info pathway_info is dataframe of the information of pathway, 539rows, 4columns.*

Description

pathway_info pathway_info is dataframe of the information of pathway, 539rows, 4columns.

Format

a list

Examples

```
data(pathway_info)
```

`pathway_info_hsa` *pathway_info pathway_info is dataframe of the information of pathway, 539rows, 4columns.*

Description

pathway_info pathway_info is dataframe of the information of pathway, 539rows, 4columns.

Format

a list pathway_info_hsa pathway_info_hsa is dataframe of the information of pathway, 539rows, 4columns. The main data is about human.

Examples

```
data(pathway_info_hsa)
```

`PFP` *The NFP package*

Description

This package implementation the applications of network finger print method.

`PFP-class` `PFP-class`

Description

An S4 object for storing pathway fingerprint scores information.

Slots

`pathways_score,` a list contains PFP_score, stats_test, genes_score. PFP_score is a numeric score indicating the performance of a gene_list in some pathways.stats_test is a statistic test for the PFP_score. genes_score is the detail scores of every gene in the gene_list.

`refnet_info,` a data.frame, which contains the specific information of pathway networks. Just be the same as `net_info` in `PFPRefnet-class`, including the index, id, name, group and species.

method

- pathways_score, `signature(object = "PFP")`: extract the pathways score
- refnet_info, `signature(object = "PFP")`: extract the pathway networks information
- PFP_score, `signature(object = "PFP")`: extract the PFP score
- stats_test, `signature(object = "PFP")`: extract p_value & p_adj_value
- genes_score, `signature(object = "PFP", index=NULL, index_type = c("pathway_id","pathway_name","slice` extract the genes score
- refnet_names, `signature(object = "PFP")`: extract the refnet names
- sub_PFP, `signature(object = "PFP", group_name = NULL, index = NULL, index_type = c("slice","pathway_i` subset of PFP object
- show_PFP, `signature(object = "PFP")`: display methods for S4 classes PFP
- plot_PFP, `signature(object, type = "character", p_size = "numeric", l_size = 'numeric')`: plot the Pathway Fingerprint.
- rank_PFP `signature(object = "PFP", total_rank = FALSE, decreasing=TRUE)` sort the PFP score.

See Also

`pathways_score-methods`, `refnet_info-methods`,`PFP_score-methods`, `stats_test-methods`, `genes_score-methods`, `refnet_names-methods`, `sub_PFP-methods`, `show_PFP-methods`, `plot_PFP-methods`, `rank_PFP-methods`,

Examples

```
data(PFP_test1)
PFP_test1
```

PFPRefnet-class PFPRefnet-class

Description

An S4 object for storing PFP reference network information.

Value

a object of PFPRefnet class

Slots

`network`, object of graphNEL list represents the basic networks.

`net_info`, a dataframe which contains the index, id, name, group and species.It contains the information of the pathway networks, whose row number is the same with *network*.

#'@section method:

- network, `signature(object = "PFPRefnet")`: extract networks of PFPRefnet
- net_info, `signature(object = "PFPRefnet")`: extract net information of PFPRefnet
- group, `signature(object = "PFPRefnet")`: extract group information
- net_names, `signature(object = "PFPRefnet")`: the names of basic networks
- subnet, `signature(object = "PFPRefnet")`: subset basic networks, e.g. a group of a networks or some networks of some given groups
- show_net, `signature(object = "PFPRefnet")`: display methods for S4 classes PFPRefnet, see also `show_net`

See Also

`network-methods`, `net_info-methods`, `group-methods`, `net_names-methods`, `subnet-methods`, `show_net-methods`,

Examples

```
data(PFPRefnet_hsa)
PFPRefnet_hsa
```

`PFPRefnet_hsa` *Pathway fingerprint data of human*

Description

A dataset containing the pathway maps of KEGG PFPRefnet_hsa is a PFPRefnet class with network(a list of length 338), net_info

Format

A PFPRefnet object, more details see `PFPRefnet-class`

See Also

`PFPRefnet-class`

Examples

```
data(PFPRefnet_hsa)
```

`PFPRefnet_mmu` *Pathway fingerprint data of mouse*

Description

A dataset containing the pathway maps of KEGG PFPRefnet_mmu a PFPRefnet class with network(a list of length 334), net_info (a datafame, 334rows and 5 columns), the main data is about mouse.

Format

A PFPRefnet object, more details see `PFPRefnet-class`

See Also

`PFPRefnet-class`

Examples

```
data(PFPRefnet_mmu)
```

`PFP_score-methods` *The score of* PFP

Description

This function can extract the PFP_score of PFP.

Usage

```
PFP_score(object)

## S4 method for signature 'PFP'
PFP_score(object)
```

Arguments

`object,`	PFP class

Value

the PFP_score

See Also

`PFP-class`

Examples

```
data(PFP_test1)
PFP_score <- PFP_score(PFP_test1)
```

`PFP_test1` *PFP_test1*

Description

A dataset of PFP class a PFP class with pathways_score(a list of length 3), refnet_info (a datafame, 338rows and 3 columns), the main data is about human.

Format

A PFPRefnet object,

See Also

`PFP-class`

Examples

```
data(PFP_test1)
```

`PFP_test2` *PFP_test2*

Description

A dataset of PFP class

Format

A PFPRefnet object,

Details

a PFP class with pathways_score(a list of length 3), refnet_info (a datafame, 338rows and 3 columns), the main data is about human.

See Also

`PFP-class`

Examples

```
data(PFP_test2)
```

`plot_PFP-methods` *Plot PFP results*

Description

Function for visualization PFP results.

Usage

```
plot_PFP(
  object,
  type = c("matchstick", "line", "point"),
  p_size = 1,
  l_size = 0.5
)

## S4 method for signature 'PFP'
plot_PFP(
  object,
  type = c("matchstick", "line", "point"),
  p_size = 1,
  l_size = 0.5
)
```

Arguments

`object,`	PFP class
`type,`	types of the visaulization of *PFP* object, 'matchstick', 'line','point'. Default is 'matchstick'.
`p_size,`	point size of plot, default is 1.
`l_size,`	line size of plot, default is 0.5.

Value

a plot of PFP

See Also

`PFP-class`

Examples

```
data(PFP_test1)
plot_PFP(PFP_test1,'line', p_size = 1, l_size = 0.5)
```

`plot_PFPlist` *Plot multiple PFPs.*

Description

Function for visualization multiple PFPs.

Usage

```
plot_PFPlist(object, l_size = 0.5)
```

Arguments

`object,`	`PFP` a list of PFP.
`l_size,`	line size of plot, default is 0.5.

Value

plot the PFP list

See Also

`PFP-class`

Examples

```
data(PFP_test1)
pfp_list <- list(a=PFP_test1)
plot_PFPlist(pfp_list)
```

`rank_PFP-methods` *rank PFPscore*

Description

rank the PFP object by the value of PFP_score.

Usage

```
rank_PFP(
  object,
  total_rank = FALSE,
  decreasing = TRUE,
  thresh_slot = "p_adj_value",
  thresh_value = 0.05
)
```

```
## S4 method for signature 'PFP'
rank_PFP(
  object,
  total_rank = FALSE,
  decreasing = TRUE,
  thresh_slot = "p_adj_value",
  thresh_value = 0.05
)
```

Arguments

`object,`	PFP class
`total_rank,`	a logical, whether to rank in total range,the default is *TRUE*
`decreasing,`	a logical, Sorting method, the default is *TRUE*
`thresh_slot,`	a character, it could be 'p_value' or 'p_adj_value', it means the threshold slot to choose for select the significant pathway. Default is 'p_adj_value'.It also could be *NULL*,it means that you don't want to select the significant pathway and you will select all pathways.
`thresh_value,`	a numeric, threshold value of 'p_value' or 'p_adjust_value' for pathway selection

Value

a ranked PFP object.

See Also

`PFP-class`

Examples

```
data(PFP_test1)
rank_PFP(PFP_test1,
        total_rank=FALSE,
        decreasing=TRUE,
        thresh_slot="p_adj_value",
        thresh_value = 0.05)
```

`refnet_info-methods` *Basic network information of* PFP *class*

Description

This function extract the detail information of reference pathway networks.

Usage

```
refnet_info(object)

## S4 method for signature 'PFP'
refnet_info(object)
```

Arguments

`object,`	PFP class

Value

detail information of reference pathway networks

See Also

`PFP-class`

Examples

```
data(PFP_test1)
refnet_info <- refnet_info(PFP_test1)
```

`refnet_names-methods` *Names of basic networks*

Description

This function extract the reference pathway network names of PFP.

Usage

```
refnet_names(object)

## S4 method for signature 'PFP'
refnet_names(object)
```

Arguments

`object,`	`PFPRefnet` class

Value

a vector contains pathway names

Examples

```
data(PFP_test1)
refnet_names <- refnet_names(PFP_test1)
```

`result_PFP-methods` *result of the PFP object.*

Description

get the result of the PFP object.

Usage

```
result_PFP(object, thresh_slot = NULL, thresh_value = 0.05)

## S4 method for signature 'PFP'
result_PFP(object, thresh_slot = NULL, thresh_value = 0.05)
```

Arguments

`object,`	PFP class
`thresh_slot,`	a character, it could be 'p_value' or 'p_adj_value', it means the threshold slot to choose for select the significant pathway. Default is *NULL*,it means that you don't want to select the significant pathway and you will select all pathways.
`thresh_value,`	a numeric, threshold value of 'p_value' or 'p_adjust_value' for pathway selection,Default is 0.05.

Value

the scores and the information of PFP object.

See Also

`PFP-class`

Examples

```
data(PFP_test1)
result_PFP(PFP_test1,
        thresh_slot="p_adj_value",
        thresh_value = 0.05)
```

`show_net` *Show an Object*

Description

show method short for PFPRefnet object

Usage

```
show_net(object)

## S4 method for signature 'PFPRefnet'
show_net(object)
```

Arguments

`object,`	`PFPRefnet` object

Value

show the network

See Also

`PFPRefnet-class`

Examples

```
data(PFPRefnet_hsa)
show_net(PFPRefnet_hsa)
```

`show_PFP` *The show_PFP generic function*

Description

Show a short summary for PFP object.

Usage

```
show_PFP(object)

## S4 method for signature 'PFP'
show_PFP(object)
```

Arguments

`object,`	`PFP` object

Value

show the PFP

Examples

```
data(PFP_test1)
show_PFP(PFP_test1)
```

`stats_test-methods` *The P value of* PFP

Description

This function can extract the result of statistical analysis

Usage

```
stats_test(object)

## S4 method for signature 'PFP'
stats_test(object)
```

Arguments

`object,`	`PFP` class

Value

Statistical test result of each pathway score

See Also

`PFP-class`

Examples

```
data(PFP_test1)
stats_test <- stats_test(PFP_test1)
```

`subnet-methods` *Subset the basic networks*

Description

Extract or Replace parts of the PFPRefnet.

Usage

```
subnet(
  object,
  group_name = NULL,
  index = NULL,
  index_type = c("slice", "pathway_id", "pathway_name")
)

## S4 method for signature 'PFPRefnet'
subnet(
  object,
  group_name = NULL,
  index = NULL,
  index_type = c("slice", "pathway_id", "pathway_name")
)
```

Arguments

`object,`	`PFPRefnet` class.
`group_name,`	character, indicating the groups to subset.
`index,`	NULL or a list contains slice/numeric, character, specifying elements to extract. This parameter' length must be the same as `group_name`. Default is *NULL*, indicating extract all the networks of a group. See *details* for more information.
`index_type,`	character, the type pf index, which could be "slice","id","name".

Details

This function help users to extract the specific networks for customized analysis, which could be of entire group networks or some part of a specific group networks.

Note, the `index` argument is only worked while the group_name argument is consideration, which means group_name is not *NULL*. And the length must be the same as `group_name`. Default is *NULL*, indicating extract the entire group basic networks.

Value

sub the network

See Also

`PFPRefnet-class`

Examples

```
data(PFPRefnet_hsa)
subnet <- subnet(PFPRefnet_hsa)
```

---

sub_PFP-methods *subset of PFP object*

Description

This function extract the subsets of PFP-class.

Usage

```
sub_PFP(
  object,
  group_name = NULL,
  index = NULL,
  index_type = c("slice", "pathway_id", "pathway_name")
)

## S4 method for signature 'PFP'
sub_PFP(
  object,
  group_name = NULL,
  index = NULL,
  index_type = c("slice", "pathway_id", "pathway_name")
)
```

Arguments

`object,`	PFP class
`group_name,`	the group name in kegg
`index,`	the index of pathway, NULL or a list contains slice/numeric, character, specifying elements to extract. This parameter' length must be the same as `group_name`. Default is *NULL*, indicating extract all the networks of a group. See *details* for more information.
`index_type,`	the index type,such as "slice","pathway_id","pathway_name"

Details

This function help users to extract the specific networks PFPscores for customized analysis, which could be of entire group PFP or some part of a specific group PFP.

Note, the `index` argument is only worked while the group_name argument is consideration, which means group_name is not *NULL*. And the length must be the same as `group_name`. Default is *NULL*, indicating extract the entire PFP.

Value

a PFP object contains just the selected elements.

See Also

`PFP-class`

Examples

```
data(PFP_test1)
PFP_test1
```

`trans_edges_id` *trans_edges_id*

Description

translate the id name in edges_data

Usage

```
trans_edges_id(
  edges_data,
  from_type = "ENTREZID",
  to_type = "SYMBOL",
  gene_info_db = NULL
)
```

Arguments

`edges_data,`	the edges_data to translate, it can be the data.frame got from `get_exp_cor_edges` or `get_asso_net`, or a data.frame contains the same colnames with them.
`from_type,`	a character,the type of gene ID, "ENSEMBL","GO","SYMBOL" and so on.
`to_type,`	a character,the type of gene ID, "ENSEMBL","GO","SYMBOL" and so on.
`gene_info_db,`	a gene

Details

Translate the id name in edges_data. Note, the `from_type` must be consistent with the genes id type in `edges_data`. The `gene_info_db` must be consistent with the species in `edges_data`

Value

the id of the edges.

Examples

```
data(PFPRefnet_hsa)
data(gene_list_hsa)
edges_kegg <- get_bg_related_kegg(gene_list_hsa,
                                  PFPRefnet=PFPRefnet_hsa,
                                  rm_duplicated = TRUE)
```

`trans_graph2PFPRefnet` *translate graph_list to PFPRefnet class*

Description

This function will translate all graphs in `graph_list` to a `PFPRefnet-class` object.

Usage

```
trans_graph2PFPRefnet(graph_list, pathway_info)
```

Arguments

`graph_list,`	a list of `graphNEL`.
`pathway_info,`	a data.frame, which contains all kegg pathways "index", "id","name","group","species"

Details

translating all graphs in `graph_list` to a `PFPRefnet-class` object. The pathway_info can be designed by yourself, but the colnames must be "index","id","name","group" and "species".

Value

a PFPRefnet

Examples

```
data(PFPRefnet_hsa)
PFPRefnet_hsa
```

`trans_xml2graph` *translate kgml files to graphNEl*

Description

This function will translate all kegg KGML files in path `file_dir`.

Usage

```
trans_xml2graph(file_dir, test_mode = FALSE)
```

Arguments

`file_dir,`	a character, refers to the file_path where kegg KGML files are stored.
`test_mode,`	please set whether to test this function.

Details

transform all KEGG KGML files downloaded by the function kegg_download() in path `file_dir` to the graphNEL object

Value

a list of `graphNEL`

Examples

```
trans_xml2graph(file_dir, test=TRUE)
```

Index