Package 'sSNAPPY'

August 18, 2022

Title Single Sample directioNAl Pathway Perturbation analYsis

Version 1.0.2

Description A single sample pathway perturbation testing method for RNA-seq data. The method propagates changes in gene expression down gene-set topologies to compute single-sample directional pathway perturbation scores that reflect potential directions of changes.Perturbation scores can be used to test significance of pathway perturbation at both individual-sample and treatment levels.

License GPL-3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

Suggests BiocManager, BiocStyle, cowplot, DT, knitr, pander, rmarkdown, spelling, stringr, testthat (>= 3.0.0), tidyverse

Config/testthat/edition 3

SystemRequirements C++11

LazyData false

Imports dplyr, magrittr, rlang, stats, plyr, purrr, BiocParallel, graphite, Rcpp, tibble, ggplot2, ggraph, igraph, reshape2, org.Hs.eg.db, SummarizedExperiment, edgeR, methods

LinkingTo Rcpp, RcppArmadillo

Depends R (>= 4.2.0)

biocViews Software, GeneExpression, GeneSetEnrichment, GeneSignaling

URL https://wenjun-liu.github.io/sSNAPPY/

VignetteBuilder knitr

BugReports https://github.com/Wenjun-Liu/sSNAPPY/issues

Language en-US

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

git_branch RELEASE_3_15

git_last_commit 0e451a9

git_last_commit_date 2022-08-02

Date/Publication 2022-08-18

Author Wenjun Liu [aut, cre] (<https://orcid.org/0000-0002-8185-3069>)

Maintainer Wenjun Liu <wenjun.liu@adelaide.edu.au>

R topics documented:

`compute_perturbation_score`

Compute Single Sample Perturbation Score

Description

Propagate weighted single sample logFCs down the pathway topologies to compute single sample perturbation scores for each pathway

Usage

```
compute_perturbation_score(weightedFC, gsTopology)
```

Arguments

`weightedFC`	A matrix of weighted single sample logFCs derived from function `weight_ss_fc()`
`gsTopology`	List of pathway topology matrices generated using function `retrieve_topology()`

Details

This function use the algorithm adopted from `SPIA` (see citation) to compute a single sample perturbation score per sample per pathway. The rownames of the weighted single sample logFC matrix and the pathway topology matrices must use the same type of gene identifier (ie. entrez ID).

Value

A list where each element is a matrix corresponding to a pathway. Each column of an element corresponds to a sample.

References

Tarca AL, Draghici S, Khatri P, Hassan SS, Mittal P, Kim JS, Kim CJ, Kusanovic JP, Romero R. A novel signaling pathway impact analysis. Bioinformatics. 2009 Jan 1;25(1):75-82.

Examples

```
#compute weighted single sample logFCs
data(metadata_example)
data(logCPM_example)
ls <- weight_ss_fc(logCPM_example, metadata = metadata_example,
factor = "patient", control = "Vehicle")


# explore all databases supported by graphite
graphite::pathwayDatabases()
gsTopology <- retrieve_topology(database = "kegg")
ssPertScore <- compute_perturbation_score(ls$logFC, gsTopology)
```

---

generate_permuted_scores

Permute sample labels to simulate null distribution of perturbation scores

---

Description

Simulate null distributions of perturbation scores for each pathway through sample permutation.

Usage

```
generate_permuted_scores(
  expreMatrix,
  numOfTreat,
  NB = 1000,
  gsTopology,
  weight,
  BPPARAM = BiocParallel::bpparam()
)

## S4 method for signature 'matrix'
generate_permuted_scores(
  expreMatrix,
  numOfTreat,
  NB = 1000,
```

```
  gsTopology,
  weight,
  BPPARAM = BiocParallel::bpparam()
)

## S4 method for signature 'data.frame'
generate_permuted_scores(
  expreMatrix,
  numOfTreat,
  NB = 1000,
  gsTopology,
  weight,
  BPPARAM = BiocParallel::bpparam()
)

## S4 method for signature 'DGEList'
generate_permuted_scores(
  expreMatrix,
  numOfTreat,
  NB = 1000,
  gsTopology,
  weight,
  BPPARAM = BiocParallel::bpparam()
)

## S4 method for signature 'SummarizedExperiment'
generate_permuted_scores(
  expreMatrix,
  numOfTreat,
  NB = 1000,
  gsTopology,
  weight,
  BPPARAM = BiocParallel::bpparam()
)
```

Arguments

`expreMatrix`	matrix and data.frame of logCPM, or DGEList/SummarizedExperiment storing gene expression counts. Feature names need to be gene entrez IDs
`numOfTreat`	Number of treatments (including control)
`NB`	Number of permutations
`gsTopology`	List of pathway topology matrices generated using function `retrieve_topology`
`weight`	A vector of gene-wise weights derived from function `weight_ss_fc`
`BPPARAM`	The parallel back-end to uses, if not specified, it is defaulted to the one returned by `BiocParallel::bpparam()`.

Details

This generate_permuted_scores function is a generic function that can deal with multiple types of inputs. It firstly randomly permute sample labels NB times to generate permuted logFCs, which are then used to compute permuted perturbation scores for each pathway. The function outputs a list that is of the same length as the list storing pathway topology matrices. Each element of the output list is for a pathway and contains a vector of permuted perturbation score of length NB. It's assumed that the permuted perturbation scores can be used to estimate the null distributions of perturbation scores.

If the input is S4 object of DGEList or SummarizedExperiment, gene expression matrix will be extracted and converted to logCPM matrix.

The default number of permutation (NB) is set to 1000. If the requested NB is larger than the maximum number of permutations possible, NB will be set to the largest number of permutations possible instead.

Value

A list where each element is a vector of perturbation scores for a pathway.

Examples

```
#compute weighted single sample logFCs
data(metadata_example)
data(logCPM_example)
ls <- weight_ss_fc(logCPM_example, metadata = metadata_example,
 factor = "patient", control = "Vehicle")

load(system.file("extdata", "gsTopology.rda", package = "sSNAPPY"))
permutedScore <- generate_permuted_scores(logCPM_example, numOfTreat = 3,
 NB = 10, gsTopology = gsTopology, weight = ls$weight)

# To see what other parallel back-end can be used:
 BiocParallel::registered()
```

---

logCPM_example *logCPM_example: Normalised logCPM of patient-derived explant models obtained from 5 ER-positive primamry breast cancer patients (GSE80098)*

Description

This data was adopted from a study by Singhal H, et al., which was published as *Genomic agonism and phenotypic antagonism between estrogen and progesterone receptors in breast cancer* in 2016. In this study, 12 primary malignant breast tissues (8PR+ and 4 PR-) were developed into patient-derived explants and treated with Vehicle, E2, E2+R5020, or R5020 for 24 or 48 hrs. Raw data for 48-hr Vehicle-, R5020-treated and E2+R5020-treated samples were retrieved from GEO

(GSE80098) and pre-processed into raw count. Filtration was sequentially performed toremove undetectable genes and the filtered counts were normalised using conditional quantile normalisation to offset effects of systematic artefacts, such as gene length and GC contents. To reduce computing time, we randomly sampled half of the genes after filtration and used their logCPM value as the example data.

Usage

```
data(logCPM_example)
```

Format

A matrix with 7672 rows and 15 columns

Source

https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE80098

`metadata_example` *metadata_example: Sample metadata for malignant breast cancer tumors PDE from 5 ER+ breast cancer patients (GSE80098)*

Description

metadata_example: Sample metadata for malignant breast cancer tumors PDE from 5 ER+ breast cancer patients (GSE80098)

Usage

```
data(metadata_example)
```

Format

A data frame with 15 rows and 4 columns

patient patient N2-3, P4-6

treatment treatment: Vehicle, E2 or E2+R5020

PR progesterone receptor status

sample sample name, corresponding to column names of the logCPM matrix

Source

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4928895/

normalise_by_permu *Normalise test perturbation scores by permutation*

Description

Normalise test perturbation scores by permutation

Usage

```
normalise_by_permu(permutedScore, testScore, pAdj_method = "fdr")
```

Arguments

permutedScore	A list. Output of generate_permuted_scores
testScore	A matrix. Output of weight_ss_fc
pAdj_method	Method for adjusting p-values for multiple comparisons. See ?p.adjust for methods available. Default to FDR.

Details

Normalise the test perturbation scores generated by weight_ss_fc through the permuted perturbation scores derived from the generate_permuted_scores function. The mean absolute deviation(MAD) and median of perturbation scores for each pathway are firstly derived from the permuted perturbation scores. The test perturbation scores are then converted to robust z-scores using MADs and medians calculated.

Value

A dataframe.

Examples

```
load(system.file("extdata", "gsTopology.rda", package = "sSNAPPY"))
ssPertScore <- compute_perturbation_score(ls$logFC, gsTopology)
permutedScore <- generate_permuted_scores(logCPM, numOfTreat = 2,
 NB = 5, gsTopology = gsTopology, weight = ls$weight)
normalisedScores <- normalise_by_permu(permutedScore, ssPertScore)

 # Load the example output
 load(system.file("extdata", "normalisedScores.rda", package = "sSNAPPY"))
 normalisedScores
```

plot_gs_network *Plot gene-set network*

Description

Plot gene-set network

Usage

```
plot_gs_network(
  normalisedScores,
  gsTopology,
  colorBy = c("robustZ", "pvalue"),
  foldGSname = TRUE,
  foldafter = 2,
  layout = "fr",
  edgeAlpha = 0.8,
  up_col = "brown3",
  down_col = "steelblue3",
  scale_edgeWidth = 10,
  scale_nodeSize = 15,
  nodeShape = 16,
  color_lg = TRUE,
  color_lg_title = NULL,
  lb_size = 3,
  lb_color = "black",
  plotIsolated = FALSE
)
```

Arguments

normalisedScores	A dataframe as described in the details section
gsTopology	List of pathway topology matrices generated using function `retrieve_topology`
colorBy	Choose to color nodes either by "robustZ" or "pvalue". A column must exist in the normalisedScores for the chosen parameter
foldGSname	logical(1). Should long gene-set names be folded into two lines
foldafter	The number of words after which gene-set names should be folded. Defaulted to 2
layout	The layout algorithm to apply
edgeAlpha	Transparency of edges
up_col	The color to label activated gene-sets. Only applicable if `colorBy` is set to be "robustZ"
down_col	The color to label inhibited gene-sets. Only applicable if `colorBy` is set to be "robustZ"

scale_edgeWidth	Parameter for scaling edge width. Defaulted to 10. Higher numbers reduce all edge width
scale_nodeSize	Parameter for scaling node size. Defaulted to 15. Higher numbers decreases all node sizes
nodeShape	The shape to use for nodes
color_lg	logical(1). Should color legend be shown
color_lg_title	Title for the color legend
lb_size	Size of node text labels
lb_color	Color of node text labels
plotIsolated	logical(1). If nodes not connected to any other node should be plotted. Default to FALSE

Value

A ggplot2 object

Examples

```
load(system.file("extdata", "gsTopology.rda", package = "sSNAPPY"))
load(system.file("extdata", "normalisedScores.rda", package = "sSNAPPY"))
#Subset pathways significantly perturbed in sample R5020_N2_48
subset <- dplyr::filter(normalisedScores, adjPvalue < 0.05, sample == "R5020_N2_48")
# Color network plot nodes by robust z-score
plot_gs_network(subset, gsTopology,
colorBy = "robustZ", layout = "dh",
color_lg_title = "Robust Z-score")

# Color network plot nodes by p-values
plot_gs_network(subset, gsTopology, layout = "dh",
colorBy = "pvalue", color_lg_title = "P-value")
```

---

retrieve_topology *Retrieve pathway topology as weighted adjacent matrix*

Description

Retrieve pathway topology matrices and convert to normalized weighted directed adjacency matrices describing gene signaling networks.

Usage

```
retrieve_topology(database, pathwayName = NULL, beta = NULL)
```

Arguments

database	See example for supported databases.
pathwayName	Optional. Subset of pathway names as a vector.
beta	Optional. A named numeric vector of weights to be assigned to each type of gene/protein relation type. See details for more information.

Details

This function takes the pathway topology information retrieved using `graphite` and convert them to normalized weighted directed adjacency matrices describing the gene signaling network, which can be directed used to compute pathway perturbation score through *SPIA* algorithm. See cited document for more details.

The beta parameter specifies weights to be assigned to each type of gene-gene interaction. It should be a named numeric vector of length 23, whose names must be: c("activation","compound","binding/association","expression "indirect","inhibition_phosphorylation","dephosphorylation_inhibition","dissociation","dephosphorylation","activation_dep "state","activation_indirect","inhibition_ubiquination","ubiquination","expression_indirect","indirect_inhibition", "repression","binding/association_phosphorylation","dissociation_phosphorylation","indirect_phosphorylation"). If unspecified, beta will be by default chosen as: c(1,0,0,1,-1,1,0,0,-1,-1,0,0,1,0,1,-1,0,1,-1,-1,0,0,0).

The converted weighted adjacent matrices will be stored in a list. We recommend users to store the returned list as a file so this step only needs to be performed once.

This function only supports and can only be used to retreive human databases as this stage.

Value

A list where each element is a matrix corresponding to a pathway

References

Tarca AL, Draghici S, Khatri P, Hassan SS, Mittal P, Kim JS, Kim CJ, Kusanovic JP, Romero R. A novel signaling pathway impact analysis. Bioinformatics. 2009 Jan 1;25(1):75-82. Sales, G., Calura, E., Cavalieri, D. et al. graphite - a Bioconductor package to convert pathway topology to gene network. BMC Bioinformatics 13, 20 (2012).

Examples

```
# explore all databases supported by graphite
graphite::pathwayDatabases()
gsTopology <- retrieve_topology(database = "kegg")
# if only interested in selected pathways, specify the pathway names in the `pathwayName` parameter
gsTopology <- retrieve_topology(database = "kegg",
pathwayName = c("Glycolysis / Gluconeogenesis",
"Citrate cycle (TCA cycle)","Pentose phosphate pathway"))
```

sSNAPPY *sSNAPPY: A package for testing directional single sample pathway perturbation*

Description

A package for testing directional single sample pathway perturbation

weight_ss_fc *Compute weighted single sample LogFCs from normalised logCPM*

Description

Compute weighted single sample logFCs for each treated samples using normalised logCPM values. Fit a lowess curve on variance of single sample logFCs ~ mean of logCPM, and use it to predict a gene-wise weight. The weighted single sample logFCs are ready for computing perturbation scores.

Usage

```
weight_ss_fc(expreMatrix, metadata = NULL, factor, control)

## S4 method for signature 'matrix'
weight_ss_fc(expreMatrix, metadata = NULL, factor, control)

## S4 method for signature 'data.frame'
weight_ss_fc(expreMatrix, metadata = NULL, factor, control)

## S4 method for signature 'DGEList'
weight_ss_fc(expreMatrix, metadata = NULL, factor, control)

## S4 method for signature 'SummarizedExperiment'
weight_ss_fc(expreMatrix, metadata = NULL, factor, control)
```

Arguments

expreMatrix	matrix and data.frame of logCPM, or DGEList/SummarizedExperiment storing gene expression counts and sample metadata. Feature names need to be gene entrez IDs, and column names need to be sample names
metadata	Sample metadata data frame as described in the details section.
factor	Factor defines how samples can be put into matching pairs (eg. patient).
control	Treatment level that is the control.

Details

This function computes weighted single sample logFCs from normalised logCPM values, used for computing single sample perturbation scores. Since genes with smaller logCPM turn to have a larger variance among single sample logFCs. A lowess curve is fitted to estimate the relationship between variance of single sample logFCs and mean of logCPM, and the relationship is used to estimate the variance of each mean logCPM value. Gene-wise weights, which are inverse of variances, are then multiplied to single sample logFCs to downweight genes with low counts. It is assumed that the genes with extremely low counts have been removed and the count matrix has been normalised prior to logCPM matrix was derived. Rownames of the matrix must be genes' entrez ID. To convert other gene identifiers to entrz ID, see example.

If a S4 object of `DGEList` or `SummarizedExperiment` is provided as input to `expreMatrix`, gene expression matrix will be extracted from it and converted to logCPM matrix. Sample metadata will also be extracted from the same S4 object unless otherwise specified.

Provided sample metadata should have the same number of rows as the number of columns in the logCPM matrix. Metadata also must have a column called "sample" storing sample names (column names of logCPM matrix), and a column called "treatment" storing treatment of each sample.The control treatment level specified by `control` parameter must exist in the treatment column.

This analysis was designed for experimental designs that include matched pairs of samples, such as when tissues collected from the same patient were treated with different treatments to study different treatment effects. Parameter `factor` tells the function how samples can be put into matching pairs. It must also be included as a column in the metadata.

Value

A list with two elements: $weight gene-wise weights; $logFC weighted single sample logFC matrix

Examples

```
# Inspect metadata data frame to make sure it has treatment, sample and patient columns
data(metadata_example)
data(logCPM_example)
length(setdiff(colnames(logCPM_example), metadata_example$sample)) == 0
ls <- weight_ss_fc(logCPM_example, metadata = metadata_example,
 factor = "patient", control = "Vehicle")
```

Index