Package: methylGSA (via r-universe)

July 12, 2024

Type Package

Title Gene Set Analysis Using the Outcome of Differential Methylation

Version 1.23.0

Description The main functions for methylGSA are methylglm and methylRRA. methylGSA implements logistic regression adjusting number of probes as a covariate. methylRRA adjusts multiple p-values of each gene by Robust Rank Aggregation. For more detailed help information, please see the vignette.

Encoding UTF-8

Imports RobustRankAggreg, ggplot2, stringr, stats, clusterProfiler, missMethyl, org.Hs.eg.db, reactome.db, BiocParallel, GO.db, AnnotationDbi, shiny, IlluminaHumanMethylation450kanno.ilmn12.hg19, IlluminaHumanMethylationEPICanno.ilm10b4.hg19

Depends R (>= 3.5)

Suggests knitr, rmarkdown, testthat, enrichplot

License GPL-2

URL https://github.com/reese3928/methylGSA

BugReports https://github.com/reese3928/methylGSA/issues

RoxygenNote 6.1.0

VignetteBuilder knitr

biocViews DNAMethylation,DifferentialMethylation,GeneSetEnrichment,Regression, GeneRegulation,Pathways

Repository https://bioc.r-universe.dev

RemoteUrl https://github.com/bioc/methylGSA

RemoteRef HEAD

RemoteSha 54e1b7fb9987916966819b086ad7104ff44d8103

Contents

`barplot` *Barplot for methylGSA analysis result*

Description

This function visualizes methylGSA analysis result by barplot.

Usage

```
barplot(res, xaxis = "Size", num = 5, colorby = "padj", title = "")
```

Arguments

`res`	A data frame which contains methylGSA analysis result.
`xaxis`	A string which specify the x-axis in the barplot. Either "Size" (number of genes in gene set) or "Count" (number of significant genes in gene set). Default is "Size". "Count" option is not available for methylglm and methylRRA(GSEA) result.
`num`	An integer. Number of gene sets to display on the barplot. Default is 5.
`colorby`	A string. Either "pvalue" or "padj". Default is "padj".
`title`	A string. Barplot title. Default is NULL.

Details

The implementation of the function is adapted from barplot function in enrichplot package.

Value

ggplot object

References

Yu G (2018). enrichplot: Visualization of Functional Enrichment Result. R package version 1.0.2, https://github.com/GuangchuangYu/enrichplot.

Examples

```
res = data.frame(ID = c("04144", "04510", "04740", "04810", "05200"),
                 Description = c("Endocytosis", "Focal adhesion",
                 "Olfactory transduction",
                 "Regulation of actin cytoskeleton", "Pathways in cancer"),
                 Size = c(201, 200, 388, 213, 326),
                 pvalue = c(0.481, 0.696, 1, 1, 1),
                 padj = 1
                 )
barplot(res)
```

`cpg.pval` *An example of user input cpg.pval*

Description

An example of user input cpg.pval

Usage

```
cpg.pval
```

Format

A named vector contains p-values of each probe tested

`CpG2Gene` *An example of user user-supplied mapping between CpGs and genes*

Description

An example of user user-supplied mapping between CpGs and genes

Usage

```
CpG2Gene
```

Format

A data frame contains mapping between CpGs and genes

getAnnot *Get CpG annotation*

Description

This function gets CpG IDs and their corresponding gene symbols.

Usage

```
getAnnot(array.type, group = "all")
```

Arguments

array.type	A string. Either "450K" or "EPIC". Default is "450K".
group	A string. "all", "body", "promoter1" or "promoter2". Default is "all". If group = "body", only CpGs on gene body will be pulled out. If group = "promoter1" or group = "promoter2", only CpGs on promoters will be pulled out. Here is the definition of "body", "promoter1" and "promoter2" according to the annotation in IlluminaHumanMethylation450kanno.ilmn12.hg19 or IlluminaHumanMethylationEPICanno.ilm10b4.hg19. • body: CpGs whose gene group correspond to "Body" or "1stExon" • promoter1: CpGs whose gene group correspond to "TSS1500" or "TSS200" • promoter2: CpGs whose gene group correspond to "TSS1500", "TSS200", "1stExon", or "5'UTR". If group = "all", all CpGs will be pulled out.

Details

The implementation of the function is modified from .flattenAnn function in missMethyl package.

Value

A data frame contains CpG IDs and gene symbols.

References

Hansen KD (2016). IlluminaHumanMethylation450kanno.ilmn12.hg19: Annotation for Illumina's 450k methylation arrays. R package version 0.6.0.

Hansen KD (2017). IlluminaHumanMethylationEPICanno.ilm10b4.hg19: Annotation for Illumina's EPIC methylation arrays. R package version 0.6.0, https://bitbucket.com/kasperdanielhansen/Illumina_EPIC.

Phipson B, Maksimovic J and Oshlack A (2015). "missMethyl: an R package for analysing methylation data from Illuminas HumanMethylation450 platform." Bioinformatics, pp. btv560.

getDescription *Get gene set description*

Description

This function gets description of gene sets.

Usage

```
getDescription(GSids, GS.type)
```

Arguments

GSids	A vector contains gene set IDs.
GS.type	A string. "GO", "KEGG", or "Reactome".

Value

A vector contains gene sets description.

References

Carlson M (2018). GO.db: A set of annotation maps describing the entire Gene Ontology. R package version 3.6.0.

Yu G, Wang L, Han Y, He Q (2012). clusterProfiler: an R package for comparing biological themes among gene clusters. OMICS: A Journal of Integrative Biology, 16(5), 284-287.

Ligtenberg W (2017). reactome.db: A set of annotation maps for reactome. R package version 1.62.0.

Examples

```
GSids = c("GO:0007389", "GO:0000978", "GO:0043062")
Description = getDescription(GSids, "GO")
head(Description)
```

getGS *Get Gene Sets*

Description

This function gets gene sets information.

Usage

```
getGS(geneids, GS.type)
```

Arguments

geneids	A vector contains all gene ids of interest. Gene ids should be gene symbol.
GS.type	A string. "GO", "KEGG", or "Reactome".

Value

A list contains all gene sets of interest and their corresponding genes.

References

Carlson M (2017). org.Hs.eg.db: Genome wide annotation for Human. R package version 3.5.0.

Ligtenberg W (2017). reactome.db: A set of annotation maps for reactome. R package version 1.62.0.

Examples

```
geneids = c("FKBP5", "NDUFA1", "STAT5B")
GO.list = getGS(geneids, "KEGG")
head(GO.list)
```

GS.list *An example of user input gene sets*

Description

An example of user input gene sets

Usage

```
GS.list
```

Format

A list contains user input gene set names and their corresponding genes

methylglm *Implement logistic regression adjusting for number of probes in enrichment analysis*

Description

This function implements logistic regression adjusting for number of probes in enrichment analysis.

Usage

```
methylglm(cpg.pval, array.type = "450K", FullAnnot = NULL,
  group = "all", GS.list = NULL, GS.idtype = "SYMBOL",
  GS.type = "GO", minsize = 100, maxsize = 500, parallel = FALSE,
  BPPARAM = bpparam())
```

Arguments

cpg.pval	A named vector containing p-values of differential methylation test. Names should be CpG IDs.
array.type	A string. Either "450K" or "EPIC". Default is "450K". This argument will be ignored if FullAnnot is provided.
FullAnnot	A data frame provided by prepareAnnot function. Default is NULL.
group	A string. "all", "body", "promoter1" or "promoter2". Default is "all". If group = "body", only CpGs on gene body will be considered in methylglm. If group = "promoter1" or group = "promoter2", only CpGs on promoters will be considered. Here is the definition of "body", "promoter1" and "promoter2" according to the annotation in IlluminaHumanMethylation450kanno.ilmn12.hg19 or IlluminaHumanMethylationEPICanno.ilm10b4.hg19. • body: CpGs whose gene group correspond to "Body" or "1stExon" • promoter1: CpGs whose gene group correspond to "TSS1500" or "TSS200" • promoter2: CpGs whose gene group correspond to "TSS1500", "TSS200", "1stExon", or "5'UTR". If group = "all", all CpGs are considered regardless of their gene group.
GS.list	A list. Default is NULL. If there is no input list, Gene Ontology is used. Entry names are gene sets names, and elements correpond to genes that gene sets contain.
GS.idtype	A string. "SYMBOL", "ENSEMBL", "ENTREZID" or "REFSEQ". Default is "SYMBOL"
GS.type	A string. "GO", "KEGG", or "Reactome". Default is "GO"
minsize	An integer. If the number of genes in a gene set is less than this integer, this gene set is not tested. Default is 100.
maxsize	An integer. If the number of genes in a gene set is greater than this integer, this gene set is not tested. Default is 500.

parallel	either TRUE or FALSE indicating whether parallel should be used. Default is FALSE
BPPARAM	an argument provided to bplapply. See register for details.

Details

The implementation of this function is modified from goglm function in GOglm package.

Value

A data frame contains gene set tests results.

References

Mi G, Di Y, Emerson S, Cumbie JS and Chang JH (2012) Length bias correction in Gene Ontology enrichment analysis using logistic regression. PLOS ONE, 7(10): e46128

Phipson, B., Maksimovic, J., and Oshlack, A. (2015). missMethyl: an R package for analysing methylation data from Illuminas HumanMethylation450 platform. Bioinformatics, btv560.

Carlson M (2017). org.Hs.eg.db: Genome wide annotation for Human. R package version 3.5.0.

Examples

```
data(CpG2Genetoy)
data(cpgtoy)
data(GSlisttoy)
GS.list = GS.list[1:10]
FullAnnot = prepareAnnot(CpG2Gene)
res = methylglm(cpg.pval = cpg.pval, FullAnnot = FullAnnot,
GS.list = GS.list, GS.idtype = "SYMBOL")
head(res)
```

---

methylgometh *Adjusting number of probes in gene set testing using gometh or gsameth in missMethyl*

Description

This function calls gometh or gsameth function in missMethyl package to adjust number of probes in gene set testing

Usage

```
methylgometh(cpg.pval, sig.cut = 0.001, topDE = NULL,
  array.type = "450K", GS.list = NULL, GS.idtype = "SYMBOL",
  GS.type = "GO", minsize = 100, maxsize = 500)
```

Arguments

cpg.pval	A named vector containing p-values of differential methylation test. Names should be CpG IDs.
sig.cut	A numeric value indicating cut-off value for significant CpG. Default is 0.001. This argument will be ignored if topDE is provided.
topDE	An integer. The top number of CpGs to be declared as significant.
array.type	A string. Either "450K" or "EPIC". Default is "450K".
GS.list	A list. Default is NULL. If there is no input list, Gene Ontology is used. Entry names are gene sets names, and elements correpond to genes that gene sets contain.
GS.idtype	A string. "SYMBOL", "ENSEMBL", "ENTREZID" or "REFSEQ". Default is "SYMBOL".
GS.type	A string. "GO", "KEGG", or "Reactome"
minsize	An integer. If the number of genes in a gene set is less than this integer, this gene set is not tested. Default is 100.
maxsize	An integer. If the number of genes in a gene set is greater than this integer, this gene set is not tested. Default is 500.

Value

A data frame contains gene set tests results.

References

Phipson, B., Maksimovic, J., and Oshlack, A. (2015). missMethyl: an R package for analysing methylation data from Illuminas HumanMethylation450 platform. Bioinformatics, btv560.

Ligtenberg W (2017). reactome.db: A set of annotation maps for reactome. R package version 1.62.0.

Carlson M (2017). org.Hs.eg.db: Genome wide annotation for Human. R package version 3.5.0.

Examples

```
## Not run:
library(IlluminaHumanMethylation450kanno.ilmn12.hg19)
data(cpgtoy)
res = methylgometh(cpg.pval = cpg.pval, sig.cut = 0.001, GS.type = "KEGG",
minsize = 200, maxsize = 205)
head(res)

## End(Not run)
```

`methylRRA` *Enrichment analysis after adjusting multiple p-values of each gene by Robust Rank Aggregation*

Description

This function implements enrichment after adjusting multiple p-values of each gene by Robust Rank Aggregation.

Usage

```
methylRRA(cpg.pval, array.type = "450K", FullAnnot = NULL,
  group = "all", method = "ORA", sig.cut = 0.05, topDE = NULL,
  GS.list = NULL, GS.idtype = "SYMBOL", GS.type = "GO",
  minsize = 100, maxsize = 500)
```

Arguments

`cpg.pval`	A named vector containing p-values of differential methylation test. Names should be CpG IDs.
`array.type`	A string. Either "450K" or "EPIC". Default is "450K". This argument will be ignored if FullAnnot is provided.
`FullAnnot`	A data frame provided by prepareAnnot function. Default is NULL.
`group`	A string. "all", "body", "promoter1" or "promoter2". Default is "all". If group = "body", only CpGs on gene body will be considered in methylRRA. If group = "promoter1" or group = "promoter2", only CpGs on promoters will be considered. Here is the definition of "body", "promoter1" and "promoter2" according to the annotation in IlluminaHumanMethylation450kanno.ilmn12.hg19 or IlluminaHumanMethylationEPICanno.ilm10b4.hg19. • body: CpGs whose gene group correspond to "Body" or "1stExon" • promoter1: CpGs whose gene group correspond to "TSS1500" or "TSS200" • promoter2: CpGs whose gene group correspond to "TSS1500", "TSS200", "1stExon", or "5'UTR". If group = "all", all CpGs are considered regardless of their gene group.
`method`	A string. "ORA" or "GSEA". Default is "ORA"
`sig.cut`	A numeric value indicating FDR cut-off for significant gene in ORA. Default is 0.05. This argument will be ignored if topDE is provided or method = "GSEA" is used.
`topDE`	An integer. The top number of genes to be declared as significant after robust rank aggregation. This argument will be ignored if method = "GSEA" is used.
`GS.list`	A list. Default is NULL. If there is no input list, Gene Ontology is used. Entry names are gene sets names, and elements correpond to genes that gene sets contain.

GS.idtype	A string. "SYMBOL", "ENSEMBL", "ENTREZID" or "REFSEQ". Default is "SYMBOL".
GS.type	A string. "GO", "KEGG", or "Reactome". Default is "GO"
minsize	An integer. If the number of genes in a gene set is less than this integer, this gene set is not tested. Default is 100.
maxsize	An integer. If the number of genes in a gene set is greater than this integer, this gene set is not tested. Default is 500.

Value

A data frame contains gene set tests results.

References

Kolde, Raivo, et al. Robust rank aggregation for gene list integration and meta-analysis. Bioinformatics 28.4 (2012): 573-580.

Phipson, B., Maksimovic, J., and Oshlack, A. (2015). missMethyl: an R package for analysing methylation data from Illuminas HumanMethylation450 platform. Bioinformatics, btv560.

Yu, Guangchuang, et al. clusterProfiler: an R package for comparing biological themes among gene clusters. Omics: a journal of integrative biology 16.5 (2012): 284-287.

Carlson M (2017). org.Hs.eg.db: Genome wide annotation for Human. R package version 3.5.0.

Examples

```
data(CpG2Genetoy)
data(cpgtoy)
data(GSlisttoy)
GS.list = GS.list[1:10]
FullAnnot = prepareAnnot(CpG2Gene)
res1 = methylRRA(cpg.pval = cpg.pval, FullAnnot = FullAnnot,
method = "ORA", GS.list = GS.list)
head(res1)
```

---

prepareAnnot *Prepare user-supplied mapping between CpGs and genes.*

Description

This function prepares CpG to gene mapping which will be used by methylRRA and methylglm.

Usage

```
prepareAnnot(CpG2Gene, geneidtype = "SYMBOL")
```

Arguments

CpG2Gene	A matrix, or a data frame or a list contains CpG to gene mapping. For a matrix or data frame, 1st column should be CpG ID and 2nd column should be gene name. For a list, entry names should be gene names, and elements correpond to CpG IDs.
geneidtype	A string. "SYMBOL", "ENSEMBL", "ENTREZID" or "REFSEQ". Default is "SYMBOL".

Value

A data frame contains ready to use CpG to gene mapping.

References

Carlson M (2017). org.Hs.eg.db: Genome wide annotation for Human. R package version 3.5.0.

Examples

```
data(CpG2Genetoy)
FullAnnot = prepareAnnot(CpG2Gene)
head(FullAnnot)
```

---

runExample *methylGSA shiny app*

Description

This is an interface for Bioconductor package methylGSA.

Usage

```
runExample(run = TRUE)
```

Arguments

run	Run the app or not. Default is TRUE

Value

The shiny app will be opened in a web browser.

Note

In order to run the app, the following R/Bioconductor packages needs to be installed properly: shinycssloaders, DT, ggplot2, IlluminaHumanMethylation450kanno.ilmn12.hg19 (if analyzing 450K array) IlluminaHumanMethylationEPICanno.ilm10b4.hg19 (if analyzing EPIC array)

Examples

```
## Please note: in this example, the argument run is set to be FALSE in
## order to pass R CMD check. However, when using the app, users are
## expected to launch the app by runExample()
runExample(FALSE)
```

Index