Package 'dmGsea'

November 13, 2025

Type Package

Title Efficient Gene Set Enrichment Analysis for DNA Methylation Data

Version 1.1.1

Description The R package dmGsea provides efficient gene set enrichment analysis specifically for DNA methylation data. It addresses key biases, including probe dependency and varying probe numbers per gene. The package supports Illumina 450K, EPIC, and mouse methylation arrays. Users can also apply it to other omics data by supplying custom probe-to-gene mapping annotations. dmGsea is flexible, fast, and well-suited for large-scale epigenomic studies.

License Artistic-2.0

biocViews GeneSetEnrichment, Pathways,DNAMethylation,Proteomics,Sequencing, CopyNumberVariation, GeneExpression, GenomicVariation, Coverage

Depends utils,stats,parallel,Matrix,SummarizedExperiment,methods,R(>= 3.5.0)

Suggests msigdbr, org.Hs.eg.db, org.Mm.eg.db, minfi, knitr, rmarkdown, GO.db, KEGGREST, testthat, IlluminaHumanMethylationEPICanno.ilm10b4.hg19, IlluminaHumanMethylation450kanno.ilmn12.hg19, BiocStyle, RUnit

Imports dqrng,AnnotationDbi,poolr,BiasedUrn,Seqinfo

VignetteBuilder knitr

Encoding UTF-8

URL https://github.com/Bioconductor/dmGsea

BugReports https://github.com/Bioconductor/dmGsea/issues

NeedsCompilation no

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

git_branch devel

git_last_commit 248aa7c

git_last_commit_date 2025-10-31

Repository Bioconductor 3.23

Date/Publication 2025-11-13

Author Zongli Xu [cre, aut] (ORCID: <https://orcid.org/0000-0002-9034-8902>),
Alison Motsinger-Reif [aut],
Liang Niu [aut]

Maintainer Zongli Xu <xuz@niehs.nih.gov>

Contents

dmGsea 2
geneID2geneName 3
getGO 3
getIlluminaAnnotation 4
getKEGG 5
getMSigDB 5
getReactome 6
gsGene 7
gsPG 9
gsProbe 10
gsRank 12

Index **14**

`dmGsea` *Efficient Gene Set Enrichment Analysis for DNA Methylation Data*

Description

The R package dmGsea is specifically designed for DNA methylation data, provides functions to perform gene set enrichment analysis while addressing probe dependency and probe number bias. The package supports Illumina 450K, EPIC, and mouse methylation arrays and can be extended to other omics data with user-provided probe-to-gene mapping annotations. It has four main functions to perform gene set enrichment analysis. gsGene: GSEA based on aggregates association signals at gene level;gsPG: GSEA using summary statistics for independent probe groups based on gene annotation; gsProbe: GSEA using probe level p-values; gsRank: Fast ranking based GSEA based on gene level statistics

Value

Functions: gsGene, gsPG, gsProbe and gsRank.

Author(s)

Zongli Xu

geneID2geneName *Extract gene names for given entrezids*

Description

Extract gene name for given entrezid.

Usage

```
geneID2geneName(gid="NULL",species="Human")
```

Arguments

gid	Specifies a vector of entrezids
species	Specifies species including "Human" and "Mouse"

Value

A list of gene names.

Author(s)

Zongli Xu

Examples

```
gname <- geneID2geneName(gid=c("672","675"),species="Human")
```

getGO *Extract GO pathway gene set*

Description

Extract GO pathway gene set.

Usage

```
getGO(subset="BP",species="Human")
```

Arguments

species	Specifies species including "Human" and "Mouse"
subset	Specifies subset of GO pathway including "BP","MF" and "CC"

Value

A list of pathways and entrezid id in each pathway.

Author(s)

Zongli Xu

Examples

```
GO <- getGO(subset="BP",species="Human")
```

`getIlluminaAnnotation` *Extract CpG-Gene corresponding table from Illuminal annotation file.*

Description

Extract CpG-Gene correspondence table from Illuminal annotation file.

Usage

```
getIlluminaAnnotation(arrayType=c("450K","EPIC"))
```

Arguments

`arrayType`	Specifies Illuminal methylation array type, including "450K" and "EPIC"

Value

A data from table for CpG-Gene correspondence relationship.

Author(s)

Zongli Xu

Examples

```
anno <- getIlluminaAnnotation(arrayType="EPIC")
```

getKEGG *Extract KEGG pathway gene set*

Description

Extract KEGG pathway gene set.

Usage

```
getKEGG(species="Human")
```

Arguments

species	Specifies species including "Human" and "Mouse"

Value

A list of pathways and entrezid id in each pathway.

Author(s)

Zongli Xu

Examples

```
kegg <- getKEGG(species="Human")
```

getMSigDB *Extract MsigDB pathway gene set*

Description

Extract MsigDB pathway gene set.

Usage

```
getMSigDB(subset="C2",species="Human")
```

Arguments

species	Specifies species including "Human" and "Mouse"
subset	Specifies subset of MsigDB pathway including "H", "C1","C2","C3","C4","C5","C6","C7" and"C8"

Value

A list of pathways and entrezid id in each pathway.

Author(s)

Zongli Xu

Examples

```
MsigDB <- getMSigDB(subset="C2",species="Human")
```

`getReactome` *Extract Reactome pathway gene set*

Description

Extract Reactome pathway gene set.

Usage

```
getReactome(species="Human")
```

Arguments

`species`	Specifies species including "Human" and "Mouse"

Value

A list of pathways and entrezid id in each pathway.

Author(s)

Zongli Xu

References

Zongli Xu, Liang Niu, OmicGsea: Efficient gene set analysis for complex Omics data. sutmitted 2024.

Examples

```
Reactome <- getReactome(species="Human")
```

gsGene *Gene set enrichment analysis based on gene level combined test P values*

Description

The function gsGene will first combine probe level P values to gene level P values, and then using threshold or ranking based methods to perform gene set enrichment analysis. If test data were provided together with test P values, correlations between probes will be adjusted in gene level P values.

Usage

```
gsGene(probe.p,Data4Cor=NULL,method="Threshold",FDRthre=0.05,nTopGene=NULL,
        GeneProbeTable=NULL,arrayType=NULL,gSetName="KEGG",
        geneSet=NULL,species="Human",combpMethod="fisher",
        combpAdjust = "nyholt",outfile="gsGene",outGenep=FALSE,
        gseaParam=1,nperm=1e4,ncore=1)
```

Arguments

probe.p	A data frame that include columns for probe name "Name" and P values "p".
Data4Cor	Methylation Data matrix (or an SummarizedExperiment object with assays(Data4Cor)$beta as the data matrix) used to generate probe.p, row as probe and column as samples. Row names should be the same with probe names in probe.p
method	Methods for testing gene set enrichment include options for "Threshold" and "Ranking".
FDRthre	False discovery rate threshold to select list of significate genes for gene set enrichment testing in threshold-based method.
nTopGene	Specifies the number of top-ranked genes based on p-value. If provided, this will override FDRthre argument to select gene list for gene set enrichment testing in threshold based method.
GeneProbeTable	A data frame for probe to gene annotation with columns for probe names "Name" and entrez gene id "entrezid"
arrayType	Specifies array type to extract GeneProbeTable, currently support DNA methylation array ("450K","EPIC"). User need to provide GeneProbeTable for other types of dataset.
gSetName	Specifies gene set names, with options including "KEGG","GO" and "MSigDB", or subsets like "GO|BP".
geneSet	User-provided gene sets, where each set contains a list of Entrez gene IDs, and the list name corresponds to the gene set name.
species	Specifies species including "Human" and "Mouse"
combpMethod	Specifies combine p-value methods including "fisher", "invchisq", "stouffer" and "tippett"

combpAdjust	Specifies method to adjust for dependence between probes including "none", "nyholt", "liji","gao" and "galwey".
outfile	Prefix of output files.
outGenep	TRUE or FALSE, if TRUE, combined gene level p values will be saved to "gsGene_genep.csv".
gseaParam	When method="Ranking", gene-level statistics -log(p) will be raised to the power of 'gseaParam' to calculate enrichment scores
nperm	Number of permutation for Ranking method.
ncore	Number of cores will be used for computation

Value

Results will be saved in files with name prefix specified by argument outfile.

Author(s)

Zongli Xu

References

Zongli Xu,Alison A. Motsinger-Reif, Liang Niu, Efficient gene set analysis for DNA methylation addressing probe dependency and bias. in review 2024.

Examples

```
if(FALSE){
kegg <- getKEGG(species="Human")
gene1 <- unique(as.vector(unlist(kegg[1:5])))
gene2 <- unique(as.vector(unlist(kegg[6:length(kegg)])))
gene1 <- rep(gene1,sample(1:10,length(gene1),replace=TRUE))
gene2 <- rep(gene2,sample(1:10,length(gene2),replace=TRUE))
p1 <- runif(length(gene1))*(1e-6)
p2 <- runif(length(gene2))
geneid <- c(gene1,gene2)
p <- c(p1,p2)
Name <- paste0("cg",1:length(p))
probe.p <- data.frame(Name=Name,p=p)
GeneProbeTable <- data.frame(Name=Name,entrezid=geneid)
dat <- matrix(runif(length(p)*100),ncol=100)
rownames(dat) <- Name
gsGene(probe.p=probe.p,Data4Cor=dat,GeneProbeTable=GeneProbeTable,
        method="Threshold",gSetName="KEGG",species="Human",ncore=1)
gsGene(probe.p=probe.p,Data4Cor=dat,GeneProbeTable=GeneProbeTable,
        method="Ranking",gSetName="KEGG",species="Human",
        outfile="geneRank",ncore=1)
gsGene(probe.p=probe.p,GeneProbeTable=GeneProbeTable,
        method="Threshold",gSetName="KEGG",species="Human",ncore=1)
}
```

gsPG *Gene set enrichment analysis based on combined probe group P values*

Description

The function gsPG will first group probes with same gene annotation and combine P values for each group, and then using noncentral hypergeometric or Monte Carlo method to perform gene set enrichment analysis. If test data were provided together with test P values, correlations between probes will be adjusted in probe group level P values.

Usage

```
gsPG(probe.p,Data4Cor=NULL,FDRthre=0.05,nTopPG=NULL,MonteCarlo=FALSE,
    GeneProbeTable=NULL,arrayType=NULL,gSetName=NULL,geneSet=NULL,
    species="Human",combpMethod="fisher",combpAdjust = "nyholt",
    outfile="gsPG",ncore=1)
```

Arguments

probe.p	A data frame that include columns for probe name "Name" and P values "p".
Data4Cor	Methylation Data matrix (or an SummarizedExperiment object with assays(Data4Cor)$beta as the data matrix)used to generate probe.p, row as probe and column as samples. Row names should be the same with probe names in probe.p
FDRthre	False discovery rate threshold to select list of significate genes for gene set enrichment testing in threshold-based method.
nTopPG	Specifies the number of top-ranked probe group based on p-value. If provided, this will override FDRthre argument to select gene list for gene set enrichment testing in threshold based method.
MonteCarlo	If TRUE, a Monte Carlo method will be used to perform geneset test, otherwise, a noncentral hypergeometric test will be used.
GeneProbeTable	A data frame for probe to gene annotation with columns for probe names "Name" and entrez gene id "entrezid"
arrayType	Specifies array type to extract GeneProbeTable, currently support DNA methylation array ("450K","EPIC"). User need to provide GeneProbeTable for other types of dataset.
gSetName	Specifies gene set names, with options including "KEGG", "GO" and "MSigDB", or subsets like "GO|BP".
geneSet	User-provided gene sets, where each set contains a list of Entrez gene IDs, and the list name corresponds to the gene set name.
species	Specifies species including "Human" and "Mouse"
combpMethod	Specifies combine p-value methods including "fisher", "invchisq", "stouffer" and "tippett"
combpAdjust	Specifies method to adjust for dependence between probes including "none","nyholt", "liji","gao" and "galwey".

outfile	Prefix of output files.
ncore	Number of cores will be used for computation

Value

Results will be saved to files with name prefix specified by outfile argument.

Author(s)

Zongli Xu

References

Zongli Xu, Alison A. Motsinger-Reif, Liang Niu, Efficient gene set analysis for DNA methylation addressing probe dependency and bias. in review 2024.

Examples

```
if(FALSE){
kegg <- getKEGG(species="Human")
gene1 <- unique(as.vector(unlist(kegg[1:5])))
gene2 <- unique(as.vector(unlist(kegg[6:length(kegg)])))
gene1 <- rep(gene1,sample(1:10,length(gene1),replace=TRUE))
gene2 <- rep(gene2,sample(1:10,length(gene2),replace=TRUE))
p1 <- runif(length(gene1))*(1e-6)
p2 <- runif(length(gene2))
geneid <- c(gene1,gene2)
p <- c(p1,p2)
Name <- paste0("cg",1:length(p))
probe.p <- data.frame(Name=Name,p=p)
GeneProbeTable <- data.frame(Name=Name,entrezid=geneid)
dat <- matrix(runif(length(p)*100),ncol=100)
rownames(dat) <- Name
gsPG(probe.p <- probe.p,Data4Cor=dat,GeneProbeTable=GeneProbeTable,
    gSetName="KEGG",species="Human",ncore=1)
gsPG(probe.p <- probe.p,Data4Cor=dat,GeneProbeTable=GeneProbeTable,
    MonteCarlo=TRUE,gSetName="KEGG",species="Human",outfile="genePG_MC",
    ncore=1)
}
```

---

gsProbe *Gene set enrichment analysis based on probe level P values*

Description

The function gsProbe will not combine P values, instead, it will adjust for the bias caused by variable number of probes per gene using a noncentral hypergeometric test in gene set enrichment analysis.

Usage

```
gsProbe(probe.p,FDRthre=0.05,nTopProbe=NULL,sigProbe=NULL,allProbe=NULL,
        GeneProbeTable=NULL,arrayType=NULL,gSetName=NULL,geneSet=NULL,
        species="Human",outfile="gsProbe",ncore=1)
```

Arguments

probe.p	A data frame that include columns for probe name "Name" and P values "p".
FDRthre	False discovery rate threshold to select list of significate genes for gene set enrichment testing in threshold-based method.
nTopProbe	Specifies the number of top-ranked probe based on p-value. If provided, this will override FDRthre argument to select gene list for gene set enrichment testing.
sigProbe	A character vector to specifies a list significant probes, this will overrid FDRthre argument to select gene list for gene set enrichment testing.
allProbe	Probe universe, use together with sigProbe argument.
GeneProbeTable	A data frame for probe to gene annotation with columns for probe names "Name" and entrez gene id "entrezid"
arrayType	Specifies array type to extract GeneProbeTable, currently support DNA methylation array ("450K","EPIC"). User need to provide GeneProbeTable for other types of dataset.
gSetName	Specifies gene set names, with options including "KEGG","GO" and "MSigDB", or subsets like "GO|BP".
geneSet	User-provided gene sets, where each set contains a list of Entrez gene IDs, and the list name corresponds to the gene set name.
species	Specifies species including "Human" and "Mouse"
outfile	Prefix of output files.
ncore	Number of cores will be used for computation

Value

Results will be saved to files with name prefix specified by outfile argument.

Author(s)

Zongli Xu

References

Zongli Xu,Alison A. Motsinger-Reif, Liang Niu, OmicGsea: Efficient Gene Set Enrichment Analysis for Complex Omics Datasets. in review 2024.

Examples

```
if(FALSE){
kegg <- getKEGG(species="Human")
gene1 <- unique(as.vector(unlist(kegg[1:5])))
gene2 <- unique(as.vector(unlist(kegg[6:length(kegg)])))
gene1 <- rep(gene1,sample(1:10,length(gene1),replace=TRUE))
gene2 <- rep(gene2,sample(1:10,length(gene2),replace=TRUE))
p1 <- runif(length(gene1))*(1e-6)
p2 <- runif(length(gene2))
geneid <- c(gene1,gene2)
p <- c(p1,p2)
Name <- paste0("cg",1:length(p))
probe.p <- data.frame(Name=Name,p=p)
GeneProbeTable <- data.frame(Name=Name,entrezid=geneid)
dat <- matrix(runif(length(p)*100),ncol=100)
rownames(dat) <- Name
gsProbe(probe.p=probe.p,GeneProbeTable=GeneProbeTable,
    gSetName="KEGG",species="Human",ncore=1)
}
```

---

gsRank *Gene set enrichment analysis based on gene level P values*

Description

Similar to GSEA and fgsea, the function gsRank will perform geneset enrichment analysis based on pre-ranked gene list (such as results from gene expression array) using ranking based method.

Usage

```
gsRank(stats,outfile="gsRank",scoreType="std",gSetName=NULL,
        geneSet=NULL,gseaParam=1,species="Human",nperm=1e4,ncore=1)
```

Arguments

stats	Named vector of gene-level test statistics, Names should be the same with names in geneSet.
outfile	Prefix of output files.
scoreType	Defines the enrichment test type, two-sided as in the original GSEA ("std"), or one-tailed tests "pos" or "neg".
gSetName	Specifies gene set names, with options including "KEGG", "GO" and "MSigDB", or subsets like "GO|BP".
geneSet	User-provided gene sets, where each set contains a list of Entrez gene IDs, and the list name corresponds to the gene set name.
gseaParam	gene-level statistics will be raised to the power of ‘gseaParam‘ to calculate enrichment scores

species	Specifies species including "Human" and "Mouse"
nperm	Specifies the number of permutations
ncore	Number of cores will be used for computation

Value

Results will be saved to files with name prefix specified by outfile argument.

Author(s)

Zongli Xu

References

Zongli Xu, Alison A. Motsinger-Reif, Liang Niu, OmicGsea: Efficient Gene Set Enrichment Analysis for Complex Omics Datasets. in review 2024.

Examples

```
if(FALSE){
kegg <- getKEGG(species="Human")
gene <- unique(as.vector(unlist(kegg)))
p <- runif(length(gene))
names(p) <- gene
stats <- -log(p)*sample(c(1,-1),length(p),replace=TRUE)

#traditional GSEA analysis, enrichment toward higher or lower end of statstics
stats <- sort(stats,decr=TRUE)
gsRank(stats=stats,gSetName="KEGG",scoreType="std",outfile="gsea9",nperm=1e5,
    ncore=1)
#enrichment of genes with higher statistics
stats <- sort(abs(stats),decr=TRUE)
gsRank(stats=stats,gSetName="KEGG",scoreType="std",outfile="gsea10",nperm=1e5,
    ncore=1)
}
```

Index

dmGsea, 2

geneID2geneName, 3
getGO, 3
getIlluminaAnnotation, 4
getKEGG, 5
getMSigDB, 5
getReactome, 6
gsGene, 7
gsPG, 9
gsProbe, 10
gsRank, 12