Package ‘GSEA’

December 16, 2019

Type Package

Title Gene set enrichment analysis among pre-defined classes and for survival data and quantitative trait of samples

Version 0.1

Description This R package conducts gene set enrichment analysis among pre-defined classes and for survival data and quantitative trait of samples. It finds BioCarta pathways, KEGG pathways, experimentally verified transcription factor targets and experimentally verified microRNA targets with statistically significant differences among pre-defined classes in the same way as the gene set comparison tool in BRB-ArrayTools. It also finds enriched pathways that are correlated with survival or a quantitative trait of the samples.

Depends R (>= 3.6.0)

Imports Biobase, GSA, bitops, Cairo

Suggests knitr, devtools, roxygen2, rmarkdown

Author BRB-ArrayTools team `<arraytools@emmes.com>`

Maintainer BRB-ArrayTools team `<arraytools@emmes.com>`

License Same as BRB-ArrayTools
(https://brb.nci.nih.gov/BRB-ArrayTools/)

Encoding UTF-8

LazyData true

RoxygenNote 7.0.0

VignetteBuilder knitr

R topics documented:

gsea *Gene set enrichment analysis*

Description

This function conducts gene set enrichment analysis among pre-defined classes and for survival data and quantitative trait data, respectively. It finds BioCarta pathways, KEGG pathways, transcription factor target lists or microRNA target lists with statistically significant differences among pre-defined classes. It aslo finds gene sets that are correlated with survival or quantitative trait of samples. A gene set is selected if its corresponding re-sampling p-value is below the specified threshold. The re-sampling p-value is calculated through permutation tests. Basically, 100,000 LS (log score) or KS (Kolmogorov-Smirnov) permutation tests are conducted to calculate a p-value measuring the gene set enrichment. For each gene set, N genes are randomly selected from a gene list in analysis, where N is the number of genes belonging to that gene set For each permutation, the LS or KS statistics associated with that gene set are computed based on the p-value. The p-value for that gene set is then defined as the proportion of permutations for which the LS (KS) statistics are larger than the observed LS (KS) statistics from original data.

Usage

```
gsea(expr, filter, surv = FALSE, time = NULL, status = NULL,
  quant = NULL, geneId, cls, isPaired = FALSE, pairID = NULL,
  doGroupComparison = FALSE, grpID = NULL, hasDuplic = FALSE,
  duplicID = NULL, rvm = TRUE, nperm.GSA = 200,
  geneSetType = c("BioCarta", "KEGG", "TF", "microRNA"),
  fromKEGGdb = TRUE, frommiRTarBasev4 = TRUE, pathwayMin = 5,
  pathwayMax = 1000, isSingleChannel = T, alpha = 0.05,
  seed = 123456, organism = c("human", "mouse"),
  corrtest.method = c("pearson", "spearman"), projectPath,
  outputName = "GeneSetClassComparison", popHTML = TRUE)
```

Arguments

`expr` matrix of gene expression data for training samples. Rows are genes and columns are arrays. Its column names must be provided.

`filter` vector of 1's or 0's of the same length as genes. 1 means to keep the gene while 0 means to exclude genes from the gene set enrichment analysis. If `rvm = TRUE`, all genes will be used in random variance model estimation.

`surv` logical specifying if it is survival data or not. Default is `FALSE`.

`time` vector specifying survival time. Defualt is `NULL`.

`status` vector specifying survival status. Defualt is `NULL`.

`quant` vector specifying the quantitative trait of samples. Default is `NULL`.

`geneId` matrix/data frame of gene IDs. Rows are IDs and columns are annotations such as Symbol and EntrezID. Its row names must be provided as IDs, and one of its column names must represent gene symbols. The rows must be in the same order as those of IDs in `expr`.

`cls` vector of sample classes.

`isPaired` logical. If `rvm` = `TRUE`, samples are paired.

`pairID` vector of pairing variables for all samples.

`doGroupComparison` logical. If `TRUE`, it compares enrichment between two groups samples. Default is `FALSE`.

`grpID` vector specifying the group ID. Default is `NULL`.

`hasDuplic` logical. If `rvm` = `TRUE`, array replicates will be averaged. Default is FALSE.

`duplicID` vector specifying array replicates. Default is `NULL`.

`rvm` logical. If `TRUE`, random variance model will be employed. Default is `TRUE`.

`nperm.GSA` numeric specifying the number of permutation tests in `GSA()` function. Defualt is 200.

`geneSetType` vector specifying the type of gene sets for enrichment analysis. It can be "BioCarta" , "KEGG", "TF" and "microRNA", representing BioCarta pathways, KEGG pathways, experimentally verified transcription factor targets and experimentally verified microRNA targets, respectively.

`fromKEGGdb` logical specifying if KEGG pathways are obtained from the KEGG.db or KEGGREST R package. When `fromKEGGdb` = `TRUE`, the 229 KEGG pathways from the KEGG.db R package are used. When `fromKEGGdb` = `FALSE`, the 333 KEGG pathways obtained from the KEGGREST R package are used. Default is `TRUE`.

`frommiRTarBasev4` logical specifying if the microRNA target lists are obtained from the miRTarBase database v4 or v7 at http://mirtarbase.mbc.nctu.edu.tw/php/download.php. When `frommiRTarBasev4` = `TRUE`, the microRNA target lists from miRTarBase v4.x are used. When `frommiRTarBasev4` = `FALSE`, the microRNA target lists from miRTarBase v7.0 are used. Default is `TRUE`.

`pathwayMin` the minimal number of genes being allowed in one pathway. Default is 5.

`pathwayMax` the maximal number of genes being allowed in one pathway. Default is 2000.

`isSingleChannel` logical. If `TRUE`, data are single-channel; otherwsie, they are two-channel. Default is `TRUE`.

`alpha` numeric specifying the significant level being set for significantly enriched pathways. Default is 0.05.

`seed` numeric specifying random seed for permutation tests. Default is 123456.

`organism` character specifying the organism. It can be "human" or "mouse". When `doBiocarta` = `TRUE`, only "human" is available for `organism`.

`corrtest.method` charcater specifying the Pearson or Spearman correlation test to find enriched gene sets given the quantitative trait data. It can be "pearson" or "spearman".

`projectPath` character specifying the ouput directory.

`outputName` character specifying the directory for keeping the HTML document and associated results. Default is "GeneSetClassComparison".

`popHTML` logical. If TRUE, an HTML document with significantly enriched pathways will be popped up. The file will be saved as <projectPath>/Output/<outputName>/<outputName>.html. Default is `TRUE`.

Value

a data frame containing the enriched BioCarta pathways, KEGG pathways, transcription factor target lists or microRNA target lists, pathway description, number of genes in each enriched pathway, p-values for LS/KS permutation tests and Efron-Tibshirani's GSA tests.

Author(s)

BRB-ArrayTools Development Team, `<arraytools@emmes.com>`

References

Xu X, Zhao Y and Simon R. Gene Set Expression Comparison kit for BRB-ArrayTools. Bioinformatics 2008. 24: 137-9.
BRB-ArrayTools manual: https://brb.nci.nih.gov/BRB-ArrayTools/Documentation.html.

Examples

```
## find BioCarta significant pathways among two classes in a breast cancer dataset
dataset<-"Brca"
# gene IDs
geneId <- read.delim(system.file("extdata", paste0(dataset, "_GENEID.txt"),
                                 package = "GSEA"), as.is = TRUE,
                                 colClasses = "character")
# gene expression
x <- read.delim(system.file("extdata", paste0(dataset, "_LOGRAT.TXT"),
                            package = "GSEA"), header = FALSE)
filter <- scan(system.file("extdata", paste0(dataset, "_FILTER.TXT"),
                           package = "GSEA"), quiet = TRUE)
# sample information
expdesign <- read.delim(system.file("extdata", paste0(dataset, "_EXPDESIGN.txt"),
                                    package = "GSEA"), as.is = TRUE)
ind1 <- which(expdesign[, 4] == "BRCA1")
ind2 <- which(expdesign[, 4] == "BRCA2")
ind <- c(ind1, ind2)
expr <- x[, ind]
colnames(expr) <- expdesign[ind, 1]
projectPath <- file.path(Sys.getenv("HOME"),"Brca")
outputName <- "GeneSetClassComparison"
cls <- c(rep("BRCA1", length(ind1)), rep("BRCA2", length(ind2)))
gsea(expr = expr,
     filter = filter,
     geneId = geneId,
     cls = cls,
     geneSetType = "BioCarta",
     isSingleChannel = FALSE,
     alpha = 0.005,
     organism = "human",
     projectPath = projectPath,
     outputName = outputName)

## find BioCarta pathways that are significantly correlated with survival
dataset<-"Pomeroy"
# gene IDs
geneId <- read.delim(system.file("extdata", paste0(dataset, "_GENEID.txt"),
                                 package = "GSEA"), as.is = TRUE,
                                 colClasses = "character")
# expression data
x <- read.delim(system.file("extdata", paste0(dataset, "_LOGINT.TXT"),
                            package = "GSEA"), header = FALSE)
# filter information, 1 - pass the filter, 0 - filtered
filter <- scan(system.file("extdata", paste0(dataset, "_FILTER.TXT"),
                           package = "GSEA"), quiet = TRUE)
# sample information
expdesign <- read.delim(system.file("extdata", paste0(dataset, "_EXPDESIGN.txt"),
                                   package = "GSEA"), as.is = TRUE)
time <- expdesign[,7]
status <- expdesign[,12]
ind1 <- which(status == 0)
ind2 <- which(status == 1)
ind <- c(ind1, ind2)
expr <- x[, ind]
time <- time[ind]
status <- status[ind]
colnames(expr) <- expdesign[ind, 1]
projectPath <- file.path(Sys.getenv("HOME"),dataset)
outputName <- "GeneSetSurvivalComparison"
gsea(expr = expr,
     filter = filter,
     surv = T,
     time = time,
     status = status,
     geneId = geneId,
     rvm = FALSE,
     geneSetType = "BioCarta",
     alpha = 0.005,
     organism = "human",
     projectPath = projectPath,
     outputName = outputName)

## find KEGG pathways that are significally correlated with simulated
## quantitative trait of samples
dataset<-"Brca"
# gene IDs
geneId <- read.delim(system.file("extdata", paste0(dataset, "_GENEID.txt"),
                                 package = "GSEA"), as.is = TRUE, colClasses = "character")
# expression data
expr <- read.delim(system.file("extdata", paste0(dataset, "_LOGRAT.TXT"),
                               package = "GSEA"), header = FALSE)
# filter information, 1 - pass the filter, 0 - filtered
```

```
filter <- scan(system.file("extdata", paste0(dataset, "_FILTER.TXT"),
                           package = "GSEA"), quiet = TRUE)
# sample information
expdesign <- read.delim(system.file("extdata", paste0(dataset, "_EXPDESIGN.txt"),
                                    package = "GSEA"), as.is = TRUE)
quant <- expdesign[,11]
colnames(expr) <- expdesign[, 1]
projectPath <- file.path(Sys.getenv("HOME"),"Brca")
outputName <- "GeneSetQTComparison"
gsea(expr = expr,
     filter = filter,
     quant = quant,
     geneId = geneId,
     geneSetType = "KEGG",
     isSingleChannel = FALSE,
     alpha = 0.005,
     organism = "human",
     corrtest.method = "pearson",
     projectPath = projectPath,
     outputName = outputName)

## find significant TF target gene lists among two classes in a breast cancer dataset
dataset<-"Brca"
# gene IDs
geneId <- read.delim(system.file("extdata", paste0(dataset, "_GENEID.txt"),
                                 package = "GSEA"), as.is = TRUE,
                                 colClasses = "character")
# gene expression
x <- read.delim(system.file("extdata", paste0(dataset, "_LOGRAT.TXT"),
                            package = "GSEA"), header = FALSE)
filter <- scan(system.file("extdata", paste0(dataset, "_FILTER.TXT"),
                           package = "GSEA"), quiet = TRUE)
# sample information
expdesign <- read.delim(system.file("extdata", paste0(dataset, "_EXPDESIGN.txt"),
                                    package = "GSEA"), as.is = TRUE)
ind1 <- which(expdesign[, 4] == "BRCA1")
ind2 <- which(expdesign[, 4] == "BRCA2")
ind <- c(ind1, ind2)
expr <- x[, ind]
colnames(expr) <- expdesign[ind, 1]
projectPath <- file.path(Sys.getenv("HOME"),"Brca")
outputName <- "GeneSetClassComparison"
cls <- c(rep("BRCA1", length(ind1)), rep("BRCA2", length(ind2)))
gsea(expr = expr,
     filter = filter,
     geneId = geneId,
     cls = cls,
     geneSetType = "TF",
     isSingleChannel = FALSE,
     alpha = 0.005,
     organism = "human",
     projectPath = projectPath,
     outputName = outputName)
```

```
## find significant microRNA target gene lists among two classes
## in a breast cancer dataset
dataset<-"Brca"
# gene IDs
geneId <- read.delim(system.file("extdata", paste0(dataset, "_GENEID.txt"),
                                 package = "GSEA"), as.is = TRUE,
                                 colClasses = "character")
# gene expression
x <- read.delim(system.file("extdata", paste0(dataset, "_LOGRAT.TXT"),
                            package = "GSEA"), header = FALSE)
filter <- scan(system.file("extdata", paste0(dataset, "_FILTER.TXT"),
                           package = "GSEA"), quiet = TRUE)
# sample information
expdesign <- read.delim(system.file("extdata", paste0(dataset, "_EXPDESIGN.txt"),
                                    package = "GSEA"), as.is = TRUE)
ind1 <- which(expdesign[, 4] == "BRCA1")
ind2 <- which(expdesign[, 4] == "BRCA2")
ind <- c(ind1, ind2)
expr <- x[, ind]
colnames(expr) <- expdesign[ind, 1]
projectPath <- file.path(Sys.getenv("HOME"),"Brca")
outputName <- "GeneSetClassComparison"
cls <- c(rep("BRCA1", length(ind1)), rep("BRCA2", length(ind2)))
gsea(expr = expr,
     filter = filter,
     geneId = geneId,
     cls = cls,
     geneSetType = "microRNA",
     isSingleChannel = FALSE,
     alpha = 0.005,
     organism = "human",
     projectPath = projectPath,
     outputName = outputName)
```

Index