Package ‘liger’

July 16, 2020

Type Package

Title Lightweight Iterative Geneset Enrichment

Version 1.1.2

Description Gene Set Enrichment Analysis (GSEA) is a computational method that determines whether an a priori defined set of genes shows statistically significant, concordant differences between two biological states. The original algorithm is detailed in Subramanian et al. with 'Java' implementations available through the Broad Institute (Subramanian et al. 2005 <doi:10.1073/pnas.0506580102>). The 'liger' package provides a lightweight R implementation of this enrichment test on a list of values (Fan et al., 2017 <doi:10.5281/zenodo.887386>). Given a list of values, such as p-values or log-fold changes derived from differential expression analysis or other analyses comparing biological states, this package enables you to test a priori defined set of genes for enrichment to enable interpretability of highly significant or high fold-change genes.

License GPL-3 | file LICENSE

LazyData TRUE

Depends R (>= 2.10)

Imports graphics, stats, Rcpp, matrixStats, parallel

LinkingTo Rcpp, RcppArmadillo

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

URL `https://github.com/JEFworks/liger`

BugReports `https://github.com/JEFworks/liger/issues`

RoxygenNote 6.1.1

NeedsCompilation yes

Author Jean Fan [aut, cre] (<https://orcid.org/0000-0002-0212-5451>),
Peter Kharchenko [aut]

Maintainer Jean Fan `<jeanfan@jhu.edu>`

Repository CRAN

Date/Publication 2020-07-16 15:10:07 UTC

R topics documented:

bulk.gsea *Bulk gene set enrichment analysis*

Description

Bulk gene set enrichment analysis

Usage

```
bulk.gsea(values, set.list, power = 1, rank = FALSE, weight = rep(1,
  length(values)), n.rand = 10000, mc.cores = 1,
  quantile.threshold = min(100/n.rand, 0.1), return.details = FALSE,
  skip.qval.estimation = FALSE)
```

Arguments

`values`	vector of values with associated gene names; values must be named, according to names appearing in set.list elements
`set.list`	list of gene sets
`power`	an exponent to control the weight of the step (default: 1)
`rank`	whether to use ranks as opposed to values (default: FALSE)
`weight`	additional weights associated with each value (default: rep(1,length(values)))
`n.rand`	number of random permutations used to assess significance (default: 1e4)
`mc.cores`	number of cores for parallel processing (default: 1)
`quantile.threshold`	threshold used (default: min(100/n.rand,0.1))
`return.details`	whether to return extended details (default: FALSE)
`skip.qval.estimation`	whether to skip q-value estimation for multiple testing (default: FALSE)

Examples

```
data("org.Hs.GO2Symbol.list")
universe <- unique(unlist(org.Hs.GO2Symbol.list))  # get universe
gs <- org.Hs.GO2Symbol.list[[1]]  # get a gene set
vals <- rnorm(length(universe), 0, 10)  # simulate values
names(vals) <- universe
vals[gs] <- rnorm(length(gs), 100, 10)
gs.list <- org.Hs.GO2Symbol.list # get gene sets
# reduce n.rand for speed
bulk.gsea(values = vals, set.list = gs.list[1:3], mc.cores = 1, n.rand=100)
```

gsea *Gene set enrichment analysis*

Description

Gene set enrichment analysis

Usage

```
gsea(values, geneset, power = 1, rank = FALSE, weight = rep(1,
  length(values)), n.rand = 10000, plot = TRUE, main = "",
  return.details = FALSE, quantile.threshold = min(100/n.rand, 0.1),
  random.seed = 1, mc.cores = 1)
```

Arguments

`values` vector of values with associated gene names; values must be named, according to names appearing in set elements

`geneset` vector of genes in the gene set

`power` an exponent to control the weight of the step (default: 1)

`rank` whether to use ranks as opposed to values (default: FALSE)

`weight` additional weights associated with each value (default: rep(1,length(values)))

`n.rand` number of random permutations used to assess significance (default: 1e4)

`plot` whether to plot (default: TRUE)

`main` plot title (default: "")

`return.details` whether to return extended details (default: FALSE)

`quantile.threshold`
threshold used (default: min(100/n.rand,0.1))

`random.seed` random seed (default: 1)

`mc.cores` number of cores for parallel processing (default: 1)

Examples

```
data("org.Hs.GO2Symbol.list")
universe <- unique(unlist(org.Hs.GO2Symbol.list))  # get universe
gs <- org.Hs.GO2Symbol.list[[1]]  # get a gene set
# fake dummy example where everything in gene set is perfectly enriched
vals <- rnorm(length(universe), 0, 10)
names(vals) <- universe
vals[gs] <- rnorm(length(gs), 100, 10)
# test obviously enriched set, reduce n.rand for speed
gsea(values=vals, geneset=gs, mc.cores=1, n.rand=100, main="GO:Random")
```

`iterative.bulk.gsea` *Iterative bulk gene set enrichment analysis*

Description

Iterative bulk gene set enrichment analysis

Usage

```
iterative.bulk.gsea(..., set.list, threshold.eval = 10, n.rand = c(100,
  1000, 10000), verbose = TRUE)
```

Arguments

`...`	arguments to be passed to `bulk.gsea`
`set.list`	list of gene sets
`threshold.eval`	threshold for applying additional permutations (default: 10)
`n.rand`	list of number of random permutations used to assess significance (default: c(1e2,1e3,1e4))
`verbose`	whether to use high verbosity level (default: TRUE)

Examples

```
data("org.Hs.GO2Symbol.list")
universe <- unique(unlist(org.Hs.GO2Symbol.list))  # get universe
gs <- org.Hs.GO2Symbol.list[[1]]  # get a gene set
vals <- rnorm(length(universe), 0, 10)  # simulate values
names(vals) <- universe
vals[gs] <- rnorm(length(gs), 100, 10)
gs.list <- org.Hs.GO2Symbol.list # get gene sets
# reduce n.rand for speed
iterative.bulk.gsea(values = vals, set.list = gs.list[1:3], mc.cores = 1, n.rand=100)
```

`liger` *liger*

Description

This package contains permutation-based gene set enrichment functionalities in R

`org.Hs.GO2Symbol.list` *Human Gene Ontology to HUGO Symbol list*

Description

Human Gene Ontology to HUGO Symbol list

Usage

```
org.Hs.GO2Symbol.list
```

Format

List with each entry as a Gene Ontology gene set

Source

`http://geneontology.org/docs/download-ontology/`

Index