Package ‘escape’

January 30, 2021

Title Easy single cell analysis platform for enrichment

Version 1.0.0

Date 2020-09-01

Description A bridging R package to facilitate gene set enrichment analysis (GSEA) in the context of single-cell RNA sequencing. Using raw count information, Seurat objects, or SingleCell-Experiment format, users can perform and visualize GSEA across individual cells.

License Apache License 2.0

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

biocViews Software, SingleCell, Classification, Annotation, GeneSetEnrichment, Sequencing, GeneSignaling, Pathways

Depends R (>= 4.0)

Imports grDevices, rlang, dplyr, factoextra, ggplot2, ggrepel, GSEABase, GSVA, SingleCellExperiment, limma, ggridges, msigdbr, stats, BiocParallel

Suggests Seurat, knitr, rmarkdown, BiocStyle, testthat, dittoSeq (>= 1.1.2)

VignetteBuilder knitr

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

git_branch RELEASE_3_12

git_last_commit ea5f684

git_last_commit_date 2020-12-02

Date/Publication 2021-01-29

Author Nick Borcherding [aut, cre],
Jared Andrews [aut]

Maintainer Nick Borcherding <`ncborch@gmail.com`>

R topics documented:

`enrichIt` *Calculate gene set enrichment scores for single-cell data*

Description

This function allows users to input both the single-cell RNA-sequencing counts and any gene set pathways either from the stored data or from other sources. The enrichment calculation itself uses the gsva R package and the poisson distribution for RNA.

Usage

```
enrichIt(obj, gene.sets = NULL, groups = 1000, cores = 2)
```

Arguments

`obj`	The count matrix, Seurat, or SingleCellExperiment object.
`gene.sets`	Gene sets from `getGeneSets` to use for the enrichment analysis.
`groups`	The number of cells to separate the enrichment calculation.
`cores`	The number of cores to use for parallelization.

Value

Data frame of normalized enrichmenet scores (NES)

Author(s)

Nick Borcherding, Jared Andrews

See Also

`getGeneSets` to collect gene sets.

Examples

```
GS <- getGeneSets(library = "H")
GS <- GS[[1]] #Reduce list size for example
seurat_ex <- suppressWarnings(Seurat::pbmc_small)
ES <- enrichIt(obj = seurat_ex, gene.sets = GS)
```

getGeneSets *Get a collection of gene sets to perform enrichment on*

Description

This function allows users to select libraries and specific gene.sets to form a GeneSetCollection that is a list of gene sets.

Usage

```
getGeneSets(species = "Homo sapiens", library = NULL, gene.sets = NULL)
```

Arguments

`species` The scientific name of the species of interest in order to get correcent gene nomenclature

`library` Individual libraries or multiple libraries to select, example: library = c("H", "C5").

`gene.sets` Select gene sets or pathways, using specific names, example: pathways = c("HALLMARK_TNFA_SI

Value

List of GeneSets in collection format

Author(s)

Nick Borcherding, Jared Andrews

Examples

```
GS <- getGeneSets(library = "H")
```

getSignificance *Perform significance testing between groups and enrichement scores.*

Description

This functions takes the enrichment scores and performs statistical testing to evaluate the difference by group selected. The function can perform 3 tests: 1) linear model based on the limma package, 2) Welch's T test, and 3) one-way ANOVA. The output includes adjusted p-values based on the Benjamini Hochberg method.

Usage

```
getSignificance(enriched, group = NULL, fit = "linear.model")
```

Arguments

enriched The output of enrichIt.

group The parameter to group for the comparison, should a column of the enriched input

fit The test used for significance, either linear.model, ANOVA, or T.test

Value

Data frame of test statistics

See Also

enrichIt for generating enrichment scores.

Examples

```
ES2 <- readRDS(url(
"https://ncborcherding.github.io/vignettes/escape_enrichment_results.rds"))
output <- getSignificance(ES2, group = "Type", fit = "linear.model")
```

masterPCAPlot *Visualize the components of the PCA analysis of the enrichment results*

Description

Graph the major gene set contributors to the pcaEnrichment.

Usage

```
masterPCAPlot(enriched, PCx, PCy, top.contribution = 10)
```

Arguments

enriched The output of enrichIt.

PCx The principal component graphed on the x-axis.

PCy The principal component graphed on the y-axis.

top.contribution The number of gene sets to graph, organized by PCA contribution.

Value

ggplot2 object sumamrizing the PCA for the enrichment scores

See Also

enrichIt for generating enrichment scores.

Examples

```
ES2 <- readRDS(url(
"https://ncborcherding.github.io/vignettes/escape_enrichment_results.rds"))
masterPCAPlot(ES2, PCx = "PC1", PCy = "PC2", top.contribution = 10)
```

pcaEnrichment *Density plot of the principal components*

Description

Density plot of the principal components

Usage

```
pcaEnrichment(
  PCAout,
  PCx,
  PCy,
  colors = c("#0348A6", "#7AC5FF", "#C6FDEC", "#FFB433", "#FF4B20"),
  contours = TRUE,
  facet = NULL
)
```

Arguments

`PCAout`	The output of `performPCA`
`PCx`	The principal component graphed on the x-axis
`PCy`	The principal component graphed on the y-axis
`colors`	The color palette for the density plot
`contours`	Binary classifier to add contours to the density plot
`facet`	A parameter to separate the graph

Value

ggplot2 object of the results of PCA for the enrichment scores

See Also

`performPCA` for generating PCA results.

Examples

```
ES2 <- readRDS(url(
"https://ncborcherding.github.io/vignettes/escape_enrichment_results.rds"))
PCA <- performPCA(enriched = ES2, groups = c("Type", "Cluster"))
pcaEnrichment(PCA, PCx = "PC1", PCy = "PC2", contours = TRUE)
```

performPCA *Calculate Principal Components for the Enrichment Scores*

Description

Using all or selected enrichment scores of individual single-cells, this function will calculate principal components using scaled values and attach to the output columns to use to graph later.

Usage

```
performPCA(enriched, groups)
```

Arguments

enriched	The output of enrichIt.
groups	The column headers to use in future graphing functions.

Value

Data frame of principal compoenents

Author(s)

Nick Borcherding

Examples

```
ES2 <- readRDS(url(
"https://ncborcherding.github.io/vignettes/escape_enrichment_results.rds"))
PCA <- performPCA(enriched = ES2, groups = c("Type", "Cluster"))
```

ridgeEnrichment *Generate a ridge plot to examine enrichment distributions*

Description

This function allows to the user to examine the distribution of enrichment across groups by generating a ridge plot.

Usage

```
ridgeEnrichment(
  enriched,
  group = "cluster",
  gene.set = NULL,
  scale.bracket = NULL,
  facet = NULL,
  add.rug = FALSE,
  colors = c("#0348A6", "#7AC5FF", "#C6FDEC", "#FFB433", "#FF4B20")
)
```

Arguments

`enriched`	The output of `enrichIt`
`group`	The parameter to group, displayed on the y-axis.
`gene.set`	The gene set to graph on the x-axis.
`scale.bracket`	This will filter the enrichment scores to remove extreme outliers. Values entered (1 or 2 numbers) will be the filtering parameter using z-scores of the selected gene.set. If only 1 value is given, a seocndary bracket is autommatically selected as the inverse of the number.
`facet`	A parameter to separate the graph.
`add.rug`	Binary classifier to add a rug plot to the x-axis.
`colors`	The color palette for the ridge plot.

Value

ggplot2 object with ridge-based distributions of selected gene.set

See Also

`enrichIt` for generating enrichment scores.

Examples

```
ES2 <- readRDS(url(
"https://ncborcherding.github.io/vignettes/escape_enrichment_results.rds"))
ridgeEnrichment(ES2, gene.set = "HALLMARK_DNA_REPAIR", group = "cluster",
facet = "Type", add.rug = TRUE)
```

`splitEnrichment` *Generate a split violin plot examine enrichment distributions*

Description

This function allows to the user to examine the distribution of enrichment across groups by generating a split violin plot.

Usage

```
splitEnrichment(
  enriched,
  x.axis = NULL,
  scale.bracket = NULL,
  split = NULL,
  gene.set = NULL,
  colors = c("#0348A6", "#7AC5FF", "#C6FDEC", "#FFB433", "#FF4B20")
)
```

Arguments

`enriched` The output of `enrichIt`

`x.axis` Optional parameter for seperation.

`scale.bracket` This will filter the enrichment scores to remove extreme outliers. Values entered (1 or 2 numbers) will be the filtering parameter using z-scores of the selected gene.set. If only 1 value is given, a seocndary bracket is autommatically selected as the inverse of the number.

`split` The parameter to split, must be binary.

`gene.set` The gene set to graph on the y-axis.

`colors` The color palette for the ridge plot.

Value

ggplot2 object violin-based distributions of selected gene.set

See Also

`enrichIt` for generating enrichment scores.

Examples

```
ES2 <- readRDS(url(
"https://ncborcherding.github.io/vignettes/escape_enrichment_results.rds"))
splitEnrichment(ES2, x.axis = "cluster", split = "Type",
gene.set = "HALLMARK_DNA_REPAIR")
```

Index