Package ‘escape’

May 29, 2024

**Title**  Easy single cell analysis platform for enrichment

**Version**  2.0.0

**Date**  2024-02-29

**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 SingleCellExperiment format, users can perform and visualize ssGSEA, GSVA, AUCell, and UCell-based enrichment calculations across individual cells.

**License**  MIT + file LICENSE

**Encoding**  UTF-8

**LazyData**  false

**RoxygenNote**  7.3.1

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

**Depends**  R (>= 4.1)

**Imports**  AUCell, BiocParallel, grDevices, dplyr, ggdist, ggplot2,
  ggpiontdensity, GSEABase, GSVA, SingleCellExperiment, ggridges,
  msigdbr, stats, reshape2, patchwork, MatrixGenerics, utils,
  SummarizedExperiment, UCell, stringr, methods, SeuratObject

**Suggests**  Seurat, hexbin, scran, knitr, rmarkdown, markdown, BiocStyle,
  RColorBrewer, rlang, spelling, testthat (>= 3.0.0), vdiffr

**VignetteBuilder**  knitr

**Language**  en-US

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

**git_branch**  RELEASE_3_19

**git_last_commit**  d314a29

**git_last_commit_date**  2024-04-30

**Repository**  Bioconductor 3.19

**Date/Publication**  2024-05-29

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

**Maintainer**  Nick Borcherding <ncborch@gmail.com>
densityEnrichment

Visualize the mean density ranking of genes across gene set

This function allows the user to examine the mean ranking within the groups across the gene set. The visualization uses the density function to display the relative position and distribution of rank.

Usage

densityEnrichment(
  input.data,
  gene.set.use = NULL,
  gene.sets = NULL,
  group.by = NULL,
  palette = "inferno"
)

Arguments

input.data The single-cell object to use.
gene.set.use Selected individual gene set.
gene.sets The gene set library to use to extract the individual gene set information from.
group.by Categorical parameter to plot along the x.axis. If input is a single-cell object the default will be cluster.
palette Colors to use in visualization - input any hcl.pals.

Value

ggplot2 object mean rank gene density across groups
**escape.gene.sets**

### Examples

```r
GS <- list(Bcells = c("MS4A1", "CD79B", "CD79A", "IGH1", "IGH2"),
            Tcells = c("CD3E", "CD3D", "CD3G", "CD7", "CD8A"))

pbmc_small <- SeuratObject::pbmc_small

densityEnrichment(pbmc_small,
                  gene.set.use = "Tcells",
                  gene.sets = GS)
```

---

**escape.gene.sets**  
**Built-In Gene Sets for escape**

### Description

A list of gene sets derived from Azizi, et al 2018 PMID: 29961579) relating to tumor immunity.

---

**escape.matrix**  
**Calculate gene set enrichment scores**

### Description

This function allows users to input both the single-cell RNA-sequencing counts and output the enrichment scores as a matrix.

### Usage

```r
escape.matrix(
    input.data,
    gene.sets = NULL,
    method = "ssGSEA",
    groups = 1000,
    min.size = 5,
    normalize = FALSE,
    make.positive = FALSE,
    BPPARAM = SerialParam(),
    ...
)
```
Arguments

- **input.data**: The count matrix, Seurat, or Single-Cell Experiment object.
- **gene.sets**: Gene sets can be a list, output from `getGeneSets`, or the built-in gene sets in the escape package `escape.gene.sets`.
- **method**: Select the method to calculate enrichment, **AUCell**, **GSVA**, **ssGSEA** or **UCell**.
- **groups**: The number of cells to separate the enrichment calculation.
- **min.size**: Minimum number of gene necessary to perform the enrichment calculation.
- **normalize**: Whether to divide the enrichment score by the number of genes **TRUE** or report unnormalized **FALSE**.
- **make.positive**: During normalization shift enrichment values to a positive range **TRUE** for downstream analysis or not **TRUE** (default). Will only be applied if **normalize** = **TRUE**.
- **BPPARAM**: A BiocParallel::bpparam() object that for parallelization.
- ... pass arguments to AUCell GSVA, ssGSEA, or UCell call

Value

- matrix of enrichment scores

Author(s)

Nick Borcherding, Jared Andrews

See Also

- [getGeneSets](#) to collect gene sets.

Examples

```r
GS <- list(Bcells = c("MS4A1", "CD79B", "CD79A", "IGH1", "IGH2"),
            Tcells = c("CD3E", "CD3D", "CD3G", "CD7", "CD8A"))
pbm_small <- SeuratObject::pbmc_small
ES <- escape.matrix(pbmc_small,
                    gene.sets = GS,
                    min.size = NULL)
```

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

geneSets(
  species = "Homo sapiens",
  library = NULL,
  subcategory = NULL,
  gene.sets = NULL
)

Arguments

species The scientific name of the species of interest in order to get correct gene nomenclature
library Individual collection(s) of gene sets, e.g. c("H", "C5"). See msigdb for all MSigDB collections.
subcategory MSigDB sub-collection abbreviation, such as CGP or BP.
gene.sets Select gene sets or pathways, using specific names, example: pathways = c("HALLMARK_TNFA_SIGNALING_VIA_NFKB").

Value

A list of gene sets from msigdb.

Author(s)

Nick Borcherding, Jared Andrews

Examples

GS <- getGeneSets(library = "H")

description

Generate a ridge plot to examine enrichment distributions

Usage

geneEnrichment(
  input.data,
  assay = NULL,
  group.by = NULL,
  gene.set = NULL,
)
color.by = "group",
order.by = NULL,
scale = FALSE,
facet.by = NULL,
palette = "inferno"
)

Arguments

input.data  Enrichment output from escape.matrix or runEscape.
assay       Name of the assay to plot if data is a single-cell object.
group.by    Categorical parameter to plot along the x.axis. If input is a single-cell object the
default will be cluster.
gene.set    Gene set to plot (on y-axis).
color.by    How the color palette applies to the graph - can be "group" for a categorical
color palette based on the group.by parameter or use the gene.set name if want-
ing to apply a gradient palette.
order.by    Method to organize the x-axis: "mean" will arrange the x-axis by the mean of
the gene.set, while "group" will arrange the x-axis by in alphanumerical order.
Using NULL will not reorder the x-axis.
scale       Visualize raw values FALSE or Z-transform enrichment values TRUE.
facet.by    Variable to facet the plot into n distinct graphs.
palette     Colors to use in visualization - input any hcl.pals.

Value

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

Examples

GS <- list(Bcells = c("MS4A1", "CD79B", "CD79A", "IGH1", "IGH2"),
            Tcells = c("CD3E", "CD3D", "CD3G", "CD7", "CD8A"))
pbmc_small <- SeuratObject::pbmc_small
pbmc_small <- runEscape(pbmc_small,
                        gene.sets = GS,
                        min.size = NULL)

geyserEnrichment(pbmc_small,
                  assay = "escape",
                  gene.set = "Tcells")
heatmapEnrichment

Generate a heatmap to visualize enrichment values

Description

This function allows to the user to examine the heatmap with the mean enrichment values by group. The heatmap will have the gene sets as rows and columns will be the grouping variable.

Usage

heatmapEnrichment(
  input.data,
  assay = NULL,
  group.by = NULL,
  gene.set.use = "all",
  cluster.rows = FALSE,
  cluster.columns = FALSE,
  scale = FALSE,
  facet.by = NULL,
  palette = "inferno"
)

Arguments

input.data [Enrichment output from escape.matrix or runEscape.]
assay [Name of the assay to plot if data is a single-cell object.]
group.by [Categorical parameter to plot along the x.axis. If input is a single-cell object the default will be cluster.]
gene.set.use [Selected gene sets to visualize. If "all", the heatmap will be generated across all gene sets.]
cluster.rows [Use Euclidean distance to order the row values.]
cluster.columns [Use Euclidean distance to order the column values.]
scale [Visualize raw values FALSE or Z-transform enrichment values TRUE.]
facet.by [Variable to facet the plot into n distinct graphs.]
palette [Colors to use in visualization - input any hcl.pals.]

Value

ggplot2 object with heatmap of mean enrichment values
**pcaEnrichment**

**Examples**

```r
GS <- list(Bcells = c("MS4A1", "CD79B", "CD79A", "IGH1", "IGH2"),
            Tcells = c("CD3E", "CD3D", "CD3G", "CD7", "CD8A"))
pbmc_small <- SeuratObject::pbmc_small
pbmc_small <- runEscape(pbmc_small,
                        gene.sets = GS,
                        min.size = NULL)

heatmapEnrichment(pbmc_small,
                   assay = "escape")
```

---

**pcaEnrichment**  
**Visualize the PCA of enrichment values**

**Description**

This function allows the user to examine the distribution of principal components run on the enrichment values.

**Usage**

```r
pcaEnrichment(
  input.data,  
  dimRed = NULL,  
  x.axis = "PC1",  
  y.axis = "PC2",  
  facet.by = NULL,  
  style = "point",  
  add.percent.contribution = TRUE,  
  display.factors = FALSE,  
  number.of.factors = 10,  
  palette = "inferno"
)
```

**Arguments**

- **input.data**  
  PCA from `performPCA`.
- **dimRed**  
  Name of the dimensional reduction to plot if data is a single-cell object.
- **x.axis**  
  Component to plot on the x.axis.
- **y.axis**  
  Component set to plot on the y.axis.
- **facet.by**  
  Variable to facet the plot into n distinct graphs.
- **style**  
  Return a "hex" bin plot or a "point"-based plot.
- **add.percent.contribution**  
  Add the relative percent of contribution of the selected components to the axis labels.
display.factors

Add an arrow overlay to show the direction and magnitude of individual gene sets on the PCA dimensions.

number.of.factors

The number of gene sets to display on the overlay.

palette

Colors to use in visualization - input any hcl.pals.

Value

ggplot2 object with PCA distribution

Examples

```r
GS <- list(Bcells = c("MS4A1", "CD79B", "CD79A", "IGH1", "IGH2"),
            Tcells = c("CD3E", "CD3D", "CD3G", "CD7", "CD8A"))
pbmc_small <- SeuratObject::pbmc_small
pbmc_small <- runEscape(pbmc_small,
                        gene.sets = GS,
                        min.size = NULL)
pbmc_small <- performPCA(pbmc_small,
                        assay = "escape")
pcaEnrichment(pbmc_small,
              x.axis = "PC1",
              y.axis = "PC2",
              dimRed = "escape.PCA")
```

performNormalization

Perform Normalization on Enrichment Data

Description

This function allows users to normalize the enrichment calculations by accounting for single-cell dropout and producing positive values for downstream differential enrichment analyses. A positive range values is useful for several downstream analyses, like differential evaluation for log2-fold change, but will alter the original enrichment values.

Usage

```r
performNormalization(
  input.data,
  assay = NULL,
  gene.sets = NULL,
  make.positive = FALSE,
  scale.factor = NULL
)
```
performPCA

**Arguments**

input.data  
Enrichment output from `escape.matrix` or `runEscape`.

assay  
Name of the assay to plot if data is a single-cell object.

gene.sets  
The gene set library to use to extract the individual gene set information from.

make.positive  
Shift enrichment values to a positive range `TRUE` for downstream analysis or `not TRUE` (default).

scale.factor  
A vector to use for normalizing enrichment scores per cell.

**Value**

Single-cell object or matrix of normalized enrichment scores

**Examples**

```r
GS <- list(Bcells = c("MS4A1", "CD79B", "CD79A", "IGH1", "IGH2"),
            Tcells = c("CD3E", "CD3D", "CD3G", "CD7", "CD8A"))
pbmc_small <- SeuratObject::pbmc_small
pbmc_small <- runEscape(pbmc_small,
                         gene.sets = GS,
                         min.size = NULL)
pbmc_small <- performNormalization(pbmc_small,
                                   assay = "escape",
                                   gene.sets = GS)
```

**performPCA**  
*Perform Principal Component Analysis on Enrichment Data*

**Description**

This function allows users to calculate the principal components for the gene set enrichment values. For single-cell data, the PCA will be stored with the dimensional reductions. If a matrix is used as input, the output is a list for further plotting. Alternatively, users can use functions for PCA calculations based on their desired workflow in lieu of using `performPCA`, but will not be compatible with downstream `pcaEnrichment` visualization.

**Usage**

```r
performPCA(
  input.data,
  assay = NULL,
  scale = TRUE,
  n.dim = 1:10,
  reduction.name = "escape.PCA",
  reduction.key = "PCA"
)
```
Arguments

- **input.data**: Enrichment output from `escape.matrix` or `runEscape`.
- **assay**: Name of the assay to plot if data is a single-cell object.
- **scale**: Standardize the enrichment value (`TRUE`) or not (`FALSE`)
- **n.dim**: The number of components to calculate.
- **reduction.name**: Name of the reduced dimensions object to add if data is a single-cell object.
- **reduction.key**: Name of the key to use with the components.

Value

single-cell object or list with PCA components to plot.

Examples

```r
GS <- list(Bcells = c("MS4A1", "CD79B", "CD79A", "IGH1", "IGH2"),
           Tcells = c("CD3E", "CD3D", "CD3G", "CD7", "CD8A"))
pbmc_small <- SeuratObject::pbmc_small
pbmc_small <- runEscape(pbmc_small,
                        gene.sets = GS,
                        min.size = NULL)

pbmc_small <- performPCA(pbmc_small,
                          assay = "escape")
```

**ridgeEnrichment** *Visualize enrichment results with a ridge plot*

Description

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

Usage

```r
ridgeEnrichment(
    input.data,
    assay = NULL,
    group.by = NULL,
    gene.set = NULL,
    color.by = "group",
    order.by = NULL,
    scale = FALSE,
    facet.by = NULL,
    add.rug = FALSE,
    palette = "inferno"
)
```
Arguments

input.data | Enrichment output from escape.matrix or runEscape.
assay      | Name of the assay to plot if data is a single-cell object.
group.by   | Categorical parameter to plot along the x.axis. If input is a single-cell object the default will be cluster.
gene.set   | Gene set to plot (on y-axis).
color.by   | How the color palette applies to the graph - can be "group" for a categorical color palette based on the group.by parameter or use the gene.set name if wanting to apply a gradient palette.
order.by   | Method to organize the x-axis: "mean" will arrange the x-axis by the mean of the gene.set, while "group" will arrange the x-axis by in alphanumerical order. Using NULL will not reorder the x-axis.
scale      | Visualize raw values FALSE or Z-transform enrichment values TRUE.
facet.by   | Variable to facet the plot into n distinct graphs.
add.rug     | Add visualization of the discrete cells along the ridge plot (TRUE).
palette    | Colors to use in visualization - input any hcl.pals.

Value

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

Examples

GS <- list(Bcells = c("MS4A1", "CD79B", "CD79A", "IGH1", "IGH2"),
               Tcells = c("CD3E", "CD3D", "CD3G", "CD7","CD8A"))
pbmc_small <- SeuratObject::pbmc_small
pbmc_small <- runEscape(pbmc_small,
                         gene.sets = GS,
                         min.size = NULL)
ridgeEnrichment(pbmc_small,
                assay = "escape",
                gene.set = "Tcells")

ridgeEnrichment(pbmc_small,
                assay = "escape",
                gene.set = "Tcells",
                color.by = "Tcells")
runEscape

Enrichment calculation for single-cell workflows

Description
Run the escape-based gene-set enrichment calculation with Seurat or SingleCellExperiment pipelines

Usage
```r
runEscape(
  input.data,
  gene.sets = NULL,
  method = "ssGSEA",
  groups = 1000,
  min.size = 5,
  normalize = FALSE,
  make.positive = FALSE,
  new.assay.name = "escape",
  BPPARAM = SerialParam(),
  ...
)
```

Arguments
- **input.data**: The count matrix, Seurat, or Single-Cell Experiment object.
- **gene.sets**: Gene sets can be a list, output from `getGeneSets`, or the built-in gene sets in the escape package `escape.gene.sets`.
- **method**: Select the method to calculate enrichment, AUCell, GSVA, ssGSEA or UCell.
- **groups**: The number of cells to separate the enrichment calculation.
- **min.size**: Minimum number of gene necessary to perform the enrichment calculation.
- **normalize**: Whether to divide the enrichment score by the number of genes `TRUE` or report unnormalized `FALSE`.
- **make.positive**: During normalization shift enrichment values to a positive range `TRUE` for downstream analysis or not `TRUE` (default). Will only be applied if `normalize = TRUE`.
- **new.assay.name**: The new name of the assay to append to the single-cell object containing the enrichment scores.
- **BPPARAM**: A BiocParallel::bpparam() object that for parallelization.
- **...**: pass arguments to AUCell GSVA, ssGSEA or UCell call

Value
Seurat or Single-Cell Experiment object with escape enrichment scores in the assay slot.
Examples

```r
GS <- list(Bcells = c("MS4A1", "CD79B", "CD79A", "IGH1", "IGH2"),
           Tcells = c("CD3E", "CD3D", "CD3G", "CD7", "CD8A"))
pbmcsmall <- SeuratObject::pbmc_small
pbmc_small <- runEscape(pbmc_small, 
                       gene.sets = GS,
                       min.size = NULL)
```

**scatterEnrichment**  Generate a density-based scatter plot

**Description**

This function allows to the user to examine the distribution of 2 gene sets along the x.axis and y.axis. The color gradient is generated using the a density estimate. See `ggpointdensity` for more information.

**Usage**

```r
scatterEnrichment(
  input.data, 
  assay = NULL, 
  x.axis = NULL, 
  y.axis = NULL, 
  scale = FALSE, 
  facet.by = NULL, 
  style = "point", 
  palette = "inferno"
)
```

**Arguments**

- `input.data` : Enrichment output from `escape.matrix` or `runEscape`
- `assay` : Name of the assay to plot if data is a single-cell object.
- `x.axis` : Gene set to plot on the x.axis.
- `y.axis` : Gene set to plot on the y.axis. `group.by` parameter or use the `gene.set` name if wanting to apply a gradient palette.
- `scale` : Visualize raw values `FALSE` or Z-transform enrichment values `TRUE`.
- `facet.by` : Variable to facet the plot into n distinct graphs.
- `style` : Return a "hex" bin plot or a "point"-based plot.
- `palette` : Colors to use in visualization - input any hcl.pals.

**Value**

`ggplot2` object with a scatter plot of selected gene sets
splitEnrichment

Examples

GS <- list(Bcells = c("MS4A1", "CD79B", "CD79A", "IGH1", "IGH2"),
            Tcells = c("CD3E", "CD3D", "CD3G", "CD7", "CD8A"))

pbmc_small <- SeuratObject::pbmc_small

pbmc_small <- runEscape(pbmc_small,
                        gene.sets = GS,
                        min.size = NULL)

scatterEnrichment(pbmc_small,
                   assay = "escape",
                   x.axis = "Tcells",
                   y.axis = "Bcells")

splitEnrichment

Visualize enrichment results with a split violin plot

Description

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

Usage

splitEnrichment(
  input.data,
  assay = NULL,
  split.by = NULL,
  group.by = NULL,
  gene.set = NULL,
  order.by = NULL,
  facet.by = NULL,
  scale = TRUE,
  palette = "inferno"
)

Arguments

input.data Enrichment output from escape.matrix or runEscape.
assay Name of the assay to plot if data is a single-cell object.
split.by Variable to form the split violin, must have 2 levels.
group.by Categorical parameter to plot along the x.axis. If input is a single-cell object the default will be cluster.
gene.set Gene set to plot (on y-axis).
order.by Method to organize the x-axis - "mean" will arrange the x-axis by the mean of the gene.set, while "group" will arrange the x-axis by in alphanumerical order. Using NULL will not reorder the x-axis.
splitEnrichment

- **facet.by** Variable to facet the plot into n distinct graphs.
- **scale** Visualize raw values **FALSE** or Z-transform enrichment values **TRUE**.
- **palette** Colors to use in visualization - input any hcl.pals.

**Value**

ggplot2 object violin-based distributions of selected gene.set

**Examples**

```r
GS <- list(Bcells = c("MS4A1", "CD79B", "CD79A", "IGH1", "IGH2"),
            Tcells = c("CD3E", "CD3D", "CD3G", "CD7", "CD8A"))
pbmcsmall <- SeuratObject::pbmc_small
pbmc_small <- runEscape(pbmc_small,
                        gene.sets = GS,
                        min.size = NULL)

splitEnrichment(pbmc_small,
                assay = "escape",
                split.by = "groups",
                gene.set = "Tcells")
```
Index

densityEnrichment, 2
escape.gene.sets, 3, 4, 13
escape.matrix, 3, 6, 7, 10–12, 14, 15
getGeneSets, 4, 4, 13
geyserEnrichment, 5
hcl.pals, 2, 6, 7, 9, 12, 14, 16
heatmapEnrichment, 7
pcaEnrichment, 8, 10
performNormalization, 9
performPCA, 8, 10, 10
ridgeEnrichment, 11
runEscape, 6, 7, 10–12, 13, 14, 15
scatterEnrichment, 14
splitEnrichment, 15