

# Package ‘escape’

May 15, 2024

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

**Version** 2.1.0

**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,  
ggpointdensity, GSEABase, GSVA, SingleCellExperiment, ggridges,  
msigdb, 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), vdiff

**VignetteBuilder** knitr

**Language** en-US

**git\_url** <https://git.bioconductor.org/packages/escape>

**git\_branch** devel

**git\_last\_commit** dbea780

**git\_last\_commit\_date** 2024-04-30

**Repository** Bioconductor 3.20

**Date/Publication** 2024-05-15

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

**Maintainer** Nick Borcharding <ncborch@gmail.com>

## Contents

densityEnrichment . . . . .	2
escape.gene.sets . . . . .	3
escape.matrix . . . . .	3
getGeneSets . . . . .	4
geyserEnrichment . . . . .	5
heatmapEnrichment . . . . .	7
pcaEnrichment . . . . .	8
performNormalization . . . . .	9
performPCA . . . . .	10
ridgeEnrichment . . . . .	11
runEscape . . . . .	13
scatterEnrichment . . . . .	14
splitEnrichment . . . . .	15
<b>Index</b>	<b>17</b>

---

densityEnrichment	<i>Visualize the mean density ranking of genes across gene set</i>
-------------------	--

---

## Description

This function allows to 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 <a href="#">hcl.pals</a> .

## Value

ggplot2 object mean rank gene density across groups

**Examples**

```

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	<i>Built-In Gene Sets for escape</i>
------------------	--------------------------------------

---

**Description**

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

---

escape.matrix	<i>Calculate gene set enrichment scores</i>
---------------	---

---

**Description**

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

**Usage**

```

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

```

**Arguments**

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

**Value**

matrix of enrichment scores

**Author(s)**

Nick Borcharding, Jared Andrews

**See Also**

[getGeneSets](#) to collect gene sets.

**Examples**

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

```
getGeneSets(
  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 <a href="#">msigdb</a> 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_IN_TCELLS"). Will only be honored if library is set, too.

**Value**

A list of gene sets from msigdb.

**Author(s)**

Nick Borcharding, Jared Andrews

**Examples**

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

---

geyserEnrichment

---

*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**

```
geyserEnrichment(
  input.data,
  assay = NULL,
  group.by = NULL,
  gene.set = NULL,
```

```

    color.by = "group",
    order.by = NULL,
    scale = FALSE,
    facet.by = NULL,
    palette = "inferno"
  )

```

## Arguments

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

## 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	<i>Generate a heatmap to visualize enrichment values</i>
-------------------	--

---

## 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

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

## Value

ggplot2 object with heatmap of mean enrichment values

## 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)

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

---

pcaEnrichment

Visualize the PCA of enrichment values

---

## Description

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

## Usage

```
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 <a href="#">performPCA</a> .
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 <b>"hex"</b> bin plot or a <b>"point"</b> -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**

```
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**

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

**Arguments**

<code>input.data</code>	Enrichment output from <a href="#">escape.matrix</a> or <a href="#">runEscape</a> .
<code>assay</code>	Name of the assay to plot if data is a single-cell object.
<code>gene.sets</code>	The gene set library to use to extract the individual gene set information from.
<code>make.positive</code>	Shift enrichment values to a positive range <b>TRUE</b> for downstream analysis or not <b>TRUE</b> (default).
<code>scale.factor</code>	A vector to use for normalizing enrichment scores per cell.

**Value**

Single-cell object or matrix of normalized enrichment scores

**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)

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**

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

**Arguments**

input.data	Enrichment output from <code>escape.matrix</code> or <code>runEscape</code> .
assay	Name of the assay to plot if data is a single-cell object.
scale	Standardize the enrichment value ( <b>TRUE</b> ) or not ( <b>FALSE</b> )
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**

```
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 to the user to examine the distribution of enrichment across groups by generating a ridge plot.

**Usage**

```
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

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

## 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**

```
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**

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

```
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	<i>Generate a density-based scatter plot</i>
-------------------	--

---

## 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

```
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 <a href="#">escape.matrix</a> or <a href="#">runEscape</a> .
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. <b>group.by</b> parameter or use the <b>gene.set</b> name if wanting to apply a gradient palette.
scale	Visualize raw values <b>FALSE</b> or Z-transform enrichment values <b>TRUE</b> .
facet.by	Variable to facet the plot into n distinct graphs.
style	Return a <b>"hex"</b> bin plot or a <b>"point"</b> -based plot.
palette	Colors to use in visualization - input any <a href="#">hcl.pals</a> .

## Value

ggplot2 object with a scatter plot of selected gene.sets

**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	<i>Visualize enrichment results with a split violin plot</i>
-----------------	--

---

**Description**

This function allows to 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 <a href="#">escape.matrix</a> or <a href="#">runEscape</a> .
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 - <b>"mean"</b> will arrange the x-axis by the mean of the gene.set, while <b>"group"</b> will arrange the x-axis by in alphanumerical order. Using <b>NULL</b> will not reorder the x-axis.

facet.by	Variable to facet the plot into n distinct graphs.
scale	Visualize raw values <b>FALSE</b> or Z-transform enrichment values <b>TRUE</b> .
palette	Colors to use in visualization - input any <a href="#">hcl.pals</a> .

**Value**

ggplot2 object violin-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)

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](#)