Package: GSABenchmark (via r-universe)

May 30, 2026

Type Package

Title Tools for benchmarking single-cell gene set analysis methods

Version 1.1.0

Description GSABenchmark is a package designed for benchmarking scRNA-seq gene set analysis (scGSA) methods. It provides both traditional and novel benchmark metrics, as well as visualization tools. Currently, GSABenchmark supports 17 scGSA methods.

License MIT + file LICENSE

Imports abdiv, CSOA, decoupleR, dplyr, escape, fabR, ggplot2, ggrepel, GSVA, hammers, henna, jaccard, lsa, Matrix, MLmetrics, methods, mltools, pagoda2, paletteer, reshape2, rlang, scLang, singscore, SiPSiC, stringr, stats, VAM, withr

Encoding UTF-8

RoxygenNote 7.3.3

Suggests AUCell, BiocStyle, knitr, qs2, ranger, rmarkdown, rpart, scater, scRNAseq, scuttle, Seurat, testthat (>= 3.0.0), UCell

biocViews Software, SingleCell, GeneSetEnrichment, GeneExpression, Visualization

VignetteBuilder knitr

URL https://github.com/andrei-stoica26/GSABenchmark

BugReports https://github.com/andrei-stoica26/GSABenchmark/issues

Config/testthat/edition 3

Config/pak/sysreqs cmake libfontconfig1-dev libfreetype6-dev git libglpk-dev make libmagick++-dev gsfonts libgit2-dev libicu-dev libpng-dev libuv1-dev libxml2-dev libssl-dev python3 libx11-dev zlib1g-dev

Repository https://bioc.r-universe.dev

Date/Publication 2026-04-28 13:06:52 UTC

RemoteUrl https://github.com/bioc/GSABenchmark

RemoteRef HEAD

RemoteSha 692fda1401e03883658e448d3250f6419e6ab947

Contents

`aggregateRankPlot` *Create an aggregate rank plot from a summary object*

Description

This function creates an aggregate rank plot from a summary object.

Usage

```
aggregateRankPlot(
  smr,
  title = NULL,
  xLab = "Method",
  sigDigits = 2,
  rankMethod = c("min", "average", "first", "last", "random", "max"),
  ...
)
```

Arguments

`smr`	Complete summary list generated with `allBenchmarkResults` or `runBenchmark`.
`title`	Plot title.
`xLab`	x axis label.
`sigDigits`	Number of significant digits used when displaying mean ranks. If `NULL`, the mean ranks will not be displayed.
`rankMethod`	Rank method. Choose between 'min', 'average', 'first', 'last', 'random' and 'max'. Default in 'min'.
`...`	Additional parameters passed to other functions.

Value

A ggplot object.

Examples

```
sPath <- system.file('extdata', 'smr.qs2', package='GSABenchmark')
smr <- qs2::qs_read(sPath)
p <- aggregateRankPlot(smr)
```

`allBenchmarkPlots` *Plot the complete list of benchmark summaries*

Description

This function plots the complete list of benchmark summaries.

Usage

```
allBenchmarkPlots(smr, titleSuffix = NULL, ...)
```

Arguments

`smr`	Complete summary list generated with `allBenchmarkResults` or `runBenchmark`.
`titleSuffix`	Suffix used to create plot titles If `NULL`, the plots will have no titles.
`...`	Additional parameters passed to other functions.

Value

A list of ggplot objects.

Examples

```
sPath <- system.file('extdata', 'smr.qs2', package='GSABenchmark')
smr <- qs2::qs_read(sPath)
plots <- allBenchmarkPlots(smr)
```

`allBenchmarkResults` *Generate all benchmark results with some precomputed arguments. This function generates all benchmark results by using precomputed values for* `normSilDF`*,* `dimMat` *and* `maxDist`*.*

Description

Generate all benchmark results with some precomputed arguments. This function generates all benchmark results by using precomputed values for `normSilDF`, `dimMat` and `maxDist`.

Usage

```
allBenchmarkResults(
  scObj,
  labelCol,
  geneSets,
  gsaMethods,
  checkLabels = TRUE,
  normSilDF = NULL,
  dimMat = NULL,
  maxDist = NULL,
  efBenchmark = NULL,
  runEFBenchmark = TRUE,
  verbose = TRUE
)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`labelCol`	The metadata column containing the ground truth annotation.
`geneSets`	A list of gene sets. If not `NULL` while `efBenchmark` is `NULL`, the efficiency benchmark will be run.
`gsaMethods`	Character vector of gene set analysis methods.
`checkLabels`	Whether to check that `geneSetNames` is a subset of the values in `labelCol`.
`normSilDF`	Data frame of normalized silhouettes.
`dimMat`	UMAP dimensionality reduction matrix of the single-cell expression object. Unused if `normSilDF` is `NULL`.
`maxDist`	Maximum UMAP distance in the single-cell expression object. Unused if `normSilDF` or `dimMat` is `NULL`.
`efBenchmark`	A list of data frames generated with efficiencyBenchmark.
`runEFBenchmark`	Whether to run efficiency benchmark.
`verbose`	Whether the output of the efficiency benchmark should be verbose. Ignored if `runEFBenchmark` is `FALSE`.

Details

This function is designed to save some computational time when multiple runs are sequentially performed for the same single-cell expression object and cell identity classes, but different choices of gene sets (so long as they still describe the same classes) or methods. If a single run is planned rather than multiple ones, `runBenchmark` is a more straightforward choice, as it takes care of generating `normSilDF` and `dimMat`.

Value

A list of benchmark results.

Examples

```
scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
scObj <- qs2::qs_read(scoPath)
gsPath <- system.file('extdata', 'geneSets.qs2', package='GSABenchmark')
geneSets <- qs2::qs_read(gsPath)
scObj <- hammers::computeSilhouette(scObj, 'label')
normSilDF <- hammers::normalizeSilhouette(scObj, 'label')
dimMat <- scLang::scPCAMat(scObj)
maxDist <- max(dist(dimMat))
smr <- allBenchmarkResults(scObj, 'label', geneSets, c('CSOA', 'Zscore'),
TRUE, normSilDF, dimMat, maxDist, runEFBenchmark=FALSE)
```

`benchmarkPlots` *Plot a list of summary data frames*

Description

This function plots a list of summary data frames with methods as rows, gene sets and the average of scores across all gene sets as columns.

Usage

```
benchmarkPlots(smr, titleSuffix = NULL, lastPlotLegendLab = "Metric", ...)
```

Arguments

`smr`	List of summary data frames, whether boundary, MCC or global.
`titleSuffix`	Suffix used to create plot titles If `NULL`, the plots will have no titles.
`lastPlotLegendLab`	The legend label of the last plot.
`...`	Additional arguments passed to `scorePlot`.

Value

A list of ggplot objects.

Examples

```
sPath <- system.file('extdata', 'smr.qs2', package='GSABenchmark')
smr <- qs2::qs_read(sPath)
plots <- benchmarkPlots(smr[[1]])
```

`corrPlots` *Create correlation plots for method results*

Description

This function creates correlation plots for method results.

Usage

```
corrPlots(
  scObj,
  smr,
  titlePrefix = "Correlation plot",
  titleInfixes = NULL,
  titleSuffix = NULL,
  ...
)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`smr`	Complete summary list generated with `allBenchmarkResults` or `runBenchmark`.
`titlePrefix`	Prefix used to create a title with `subplotTitle`. Ignored if `titleSuffix` is `NULL`.
`titleInfixes`	A character vector of title infixes used to create a title with `subplotTitle`. Ignored if `titleSuffix` is `NULL`.
`titleSuffix`	Suffix used to create plot titles If `NULL`, the plots will have no titles.
`...`	Additional arguments passed to `henna::correlationPlot`.

Value

A named list of ggplot objects.

Examples

```
scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
scObj <- qs2::qs_read(scoPath)
sPath <- system.file('extdata', 'smr.qs2', package='GSABenchmark')
smr <- qs2::qs_read(sPath)
plots <- corrPlots(scObj, smr)
```

`corrSummary` *Calculate correlation matrix for method results*

Description

This function calculates the correlation matrix for all the methods for an identity class.

Usage

```
corrSummary(scObj, smr, corMethod = "spearman")
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`smr`	List containing boundary, MCC and global lists of data frames.
`corMethod`	Correlation method.

Value

A list of correlation matrices.

Examples

```
scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
scObj <- qs2::qs_read(scoPath)
sPath <- system.file('extdata', 'smr.qs2', package='GSABenchmark')
smr <- qs2::qs_read(sPath)
m <- corrSummary(scObj, smr)
```

`efficiencyBenchmark` *Extract running times and peak memory usage for gene set analysis methods*

Description

This function extracts running times and peak memory usage for gene set analysis methods.

Usage

```
efficiencyBenchmark(
  scObj,
  labelCol,
  geneSets,
  gsaMethods,
  checkLabels = TRUE,
  verbose = TRUE
)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`labelCol`	The metadata column containing the ground truth annotation.
`geneSets`	A list of gene sets. If not `NULL` while `efBenchmark` is `NULL`, the efficiency benchmark will be run.
`gsaMethods`	Character vector of gene set analysis methods.
`checkLabels`	Whether to check that `geneSetNames` is a subset of the values in `labelCol`.
`verbose`	Whether output should be verbose.

Value

A list of two data frames, the first comprising running times in seconds, the second comprising peak memory usage in mebibytes.

Examples

```
scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
scObj <- qs2::qs_read(scoPath)
gsPath <- system.file('extdata', 'geneSets.qs2', package='GSABenchmark')
geneSets <- qs2::qs_read(gsPath)
ef <- efficiencyBenchmark(scObj, 'label', geneSets[1], 'CSOA',
verbose=FALSE)
```

`geneSetRankPlots` *Create gene set rank plots for the method results*

Description

This function creates gene set rank plots for method results.

Usage

```
geneSetRankPlots(
  smr,
  rankMethod = "min",
  titlePrefix = "Distribution of ranks",
  titleInfixes = NULL,
  titleSuffix = NULL,
  xLab = "Method",
  ...
)
```

Arguments

`smr`	Complete summary list generated with `allBenchmarkResults` or `runBenchmark`.
`rankMethod`	Rank method. Choose between 'min', 'average', 'first', 'last', 'random' and 'max'. Default in 'min'.
`titlePrefix`	Prefix used to create a title with `subplotTitle`. Ignored if `titleSuffix` is `NULL`.
`titleInfixes`	A character vector of title infixes used to create a title with `subplotTitle`. Ignored if `titleSuffix` is `NULL`.
`titleSuffix`	Suffix used to create plot titles If `NULL`, the plots will have no titles.
`xLab`	x axis label.
`...`	Additional arguments passed to `henna::rankPlot`.

Value

A named list of ggplot objects.

Examples

```
sPath <- system.file('extdata', 'smr.qs2', package='GSABenchmark')
smr <- qs2::qs_read(sPath)
plots <- geneSetRankPlots(smr)
```

`mdsPlots` *Create MDS plots for method results*

Description

This function creates MDS plots for method results.

Usage

```
mdsPlots(
  scObj,
  smr,
  titlePrefix = "MDS plot",
  titleInfixes = NULL,
  titleSuffix = NULL,
  drawScores = TRUE,
  ...
)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`smr`	Complete summary list generated with `allBenchmarkResults` or `runBenchmark`.
`titlePrefix`	Prefix used to create a title with `subplotTitle`. Ignored if `titleSuffix` is `NULL`.
`titleInfixes`	A character vector of title infixes used to create a title with `subplotTitle`. Ignored if `titleSuffix` is `NULL`.
`titleSuffix`	Suffix used to create plot titles If `NULL`, the plots will have no titles.
`drawScores`	Whether to render scores on the plot. If set to `TRUE`, the third column of the input data frame will be numeric and scores will be taken from there.
`...`	Additional arguments passed to `henna::densityPlot`.

Value

A named list of ggplot objects.

Examples

```
scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
scObj <- qs2::qs_read(scoPath)
sPath <- system.file('extdata', 'smr.qs2', package='GSABenchmark')
smr <- qs2::qs_read(sPath)
plots <- mdsPlots(scObj, smr)
```

`memoryPlot` *Plot a data frame consisting of gene set analysis method peak memory usage This function plots data frame consisting of method peak memory usages with methods as rows, gene sets and the gene set average as columns.*

Description

Plot a data frame consisting of gene set analysis method peak memory usage This function plots data frame consisting of method peak memory usages with methods as rows, gene sets and the gene set average as columns.

Usage

```
memoryPlot(efBenchmark, title = NULL, xLab = "Peak memory usage (MiB)", ...)
```

Arguments

`efBenchmark`	A list of dataframes generated with `efficiencyBenchmark`.
`title`	Plot title.
`xLab`	x axis label.
`...`	Additional parameters to be passed to `scorePlot`.

Value

A ggplot object.

Examples

```
sPath <- system.file('extdata', 'smr.qs2', package='GSABenchmark')
smr <- qs2::qs_read(sPath)
memoryPlot(smr[[5]])
```

`metricRankPlots` *Create metric rank plots for the method results*

Description

This function creates metric rank plots for method results.

Usage

```
metricRankPlots(
  smr,
  titlePrefix = "Distribution of ranks",
  titleInfixes = metricNames(),
  titleSuffix = NULL,
  rankMethod = "min",
  xLab = "Method",
  ...
)
```

Arguments

`smr`	Complete summary list generated with `allBenchmarkResults` or `runBenchmark`.
`titlePrefix`	Prefix used to create a title with `subplotTitle`. Ignored if `titleSuffix` is `NULL`.
`titleInfixes`	A character vector of title infixes used to create a title with `subplotTitle`. Ignored if `titleSuffix` is `NULL`.
`titleSuffix`	Suffix used to create plot titles If `NULL`, the plots will have no titles.
`rankMethod`	Rank method. Choose between 'min', 'average', 'first', 'last', 'random' and 'max'. Default in 'min'.
`xLab`	x axis label.
`...`	Additional arguments passed to `henna::rankPlot`.

Value

A named list of ggplot objects.

Examples

```
sPath <- system.file('extdata', 'smr.qs2', package='GSABenchmark')
smr <- qs2::qs_read(sPath)
plots <- metricRankPlots(smr)
```

`predJaccardPlots` *Create Jaccard tile plots for method binary predictions*

Description

This function creates Jaccard tile plots for method binary predictions.

Usage

```
predJaccardPlots(
  predictionsSmr,
  titlePrefix = "Binary prediction Jaccard plot",
  titleInfixes = NULL,
  titleSuffix = NULL,
  ...
)
```

Arguments

`predictionsSmr`	Binary predictions summary.
`titlePrefix`	Prefix used to create a title with `subplotTitle`. Ignored if `titleSuffix` is `NULL`.
`titleInfixes`	A character vector of title infixes used to create a title with `subplotTitle`. Ignored if `titleSuffix` is `NULL`.
`titleSuffix`	Suffix used to create plot titles If `NULL`, the plots will have no titles.
`...`	Additional arguments passed to `henna::tilePlot`.

Value

A named list of ggplot objects.

Examples

```
sPath <- system.file('extdata', 'smr.qs2', package='GSABenchmark')
smr <- qs2::qs_read(sPath)
plots <- predJaccardPlots(smr$predictions)
```

`ratioPlot` *Create a ratio rank plot for the method results*

Description

This function creates a ratio rank plot for the method results.

Usage

```
ratioPlot(
  smr,
  title = NULL,
  nItems = 25,
  xLab = "Maximum over mean ratio",
  yLab = "Metric",
  legendLab = "Method",
  ...
)
```

Arguments

`smr`	Complete summary list generated with `allBenchmarkResults` or `runBenchmark`.
`title`	Plot title.
`nItems`	Number of retained iterms.
`xLab`	x axis label.
`yLab`	y axis label.
`legendLab`	Legend label.
`...`	Additional arguments passed to `henna::classRank`.

Value

A ggplot object.

Examples

```
sPath <- system.file('extdata', 'smr.qs2', package='GSABenchmark')
smr <- qs2::qs_read(sPath)
p <- ratioPlot(smr)
```

`runAddModuleScore` *Run AddModuleScore*

Description

This function runs `AddModuleScore`.

Usage

```
runAddModuleScore(
  scObj,
  geneSets,
  slot = "data",
  pool = rownames(scObj),
  nbin = 24,
  ctrl = 100,
  seed = 1
)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`geneSets`	A named list of gene sets.
`slot`	Gene expression slot.
`pool`	The set from which features to be compared with signature genes are selected. Defaults to all features.
`nbin`	Number of bins of aggregate expression levels for pool features.
`ctrl`	Number of control features chosen from the same bin for each feature.
`seed`	Random seed.

Details

Reimplemented from `https://github.com/satijalab`.

Value

A single-cell expression object with the results saved as a metadata column.

Examples

```
scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
scObj <- qs2::qs_read(scoPath)
gsPath <- system.file('extdata', 'geneSets.qs2', package='GSABenchmark')
geneSets <- qs2::qs_read(gsPath)
scObj <- runAddModuleScore(scObj, geneSets)
```

`runAUCell` *Run AUCell*

Description

This function runs `AUCell` using `escape`.

Usage

```
runAUCell(scObj, geneSets, slot = "data", ...)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`geneSets`	A named list of gene sets.
`slot`	Gene expression slot.
`...`	Additional parameters passed to `runEscapeMethod`.

Value

A single-cell expression object with the results saved as a metadata column.

Examples

```
if (requireNamespace("AUCell", quietly=TRUE)){
    scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
    scObj <- qs2::qs_read(scoPath)
    gsPath <- system.file('extdata', 'geneSets.qs2', package='GSABenchmark')
    geneSets <- qs2::qs_read(gsPath)
    scObj <- runAUCell(scObj, geneSets)
    }
```

`runBenchmark` *Generate all benchmark results This function performs the entire* `GSABenchmark` *pipeline.*

Description

Generate all benchmark results This function performs the entire `GSABenchmark` pipeline.

Usage

```
runBenchmark(
  scObj,
  labelCol,
  geneSets,
  gsaMethods,
  runEFBenchmark = TRUE,
  verbose = TRUE
)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`labelCol`	The metadata column containing the ground truth annotation.
`geneSets`	A list of gene sets. If not `NULL` while `efBenchmark` is `NULL`, the efficiency benchmark will be run.
`gsaMethods`	Character vector of gene set analysis methods.
`runEFBenchmark`	Whether to run efficiency benchmark.
`verbose`	Whether the output of the efficiency benchmark should be verbose. Ignored if `runEFBenchmark` is `FALSE`.

Details

A wrapper around `allBenchmarkResults`. Slower for repeated runs, but it does not require users to manually generate `normSilDF` and `dimMat`.

Value

A list of benchmark results.

Examples

```
scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
scObj <- qs2::qs_read(scoPath)
gsPath <- system.file('extdata', 'geneSets.qs2', package='GSABenchmark')
geneSets <- qs2::qs_read(gsPath)
smr <- runBenchmark(scObj, 'label', geneSets, c('CSOA', 'Zscore'), FALSE)
```

`runBenchmarkShuffle` *Generate all benchmark results for shuffled gene sets This function generates all benchmark results for shuffled gene sets.*

Description

Generate all benchmark results for shuffled gene sets This function generates all benchmark results for shuffled gene sets.

Usage

```
runBenchmarkShuffle(
  scObj,
  labelCol,
  geneSets,
  gsaMethod,
  runEFBenchmark = TRUE
)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`labelCol`	The metadata column containing the ground truth annotation.
`geneSets`	A list of gene sets. If not `NULL` while `efBenchmark` is `NULL`, the efficiency benchmark will be run.
`gsaMethod`	Name of the gene set analysis method.
`runEFBenchmark`	Whether to run efficiency benchmark.

Value

A list of benchmark results.

Examples

```
scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
scObj <- qs2::qs_read(scoPath)
gsPath <- system.file('extdata', 'geneSets.qs2', package='GSABenchmark')
geneSets <- qs2::qs_read(gsPath)
scObj <- runMethodShuffle(scObj, 'label', geneSets, 'CSOA', 0.2, 0.2)
smr <- runBenchmarkShuffle(scObj, 'label', geneSets, 'CSOA', FALSE)
```

`runGSAMethods` *Run gene set analysis methods*

Description

This function runs the gene set analysis methods.

Usage

```
runGSAMethods(
  scObj,
  labelCol,
  geneSets,
  gsaMethods,
```

```
  infix = NULL,
  outputFun = silently_run
)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`labelCol`	The metadata column containing the ground truth annotation.
`geneSets`	A named list of gene sets.
`gsaMethods`	Character vector of gene set analysis methods.
`infix`	Infix to add between method name and gene set name in single-cell expression object. The string consisting of the method name and the infix is separated by the gene set name with a '_' character.
`outputFun`	Choose between silently_run (suppress all warnings and messages) or identity (do not suppress them). Default is silently_run.

Value

A `Seurat` or `SingleCellExpression` object with the results of the runs stored as metadata columns.

Examples

```
scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
scObj <- qs2::qs_read(scoPath)
gsPath <- system.file('extdata', 'geneSets.qs2', package='GSABenchmark')
geneSets <- qs2::qs_read(gsPath)
scObj <- runGSAMethods(scObj, 'label', geneSets, c('CSOA', 'Zscore'))
```

`runGSVA` *Run GSVA*

Description

This function runs `GSVA` using `GSVA`.

Usage

```
runGSVA(scObj, geneSets, slot = "data", ...)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`geneSets`	A named list of gene sets.
`slot`	Gene expression slot.
`...`	Additional parameters passed to `runGSVAMethod`.

Value

A single-cell expression object with the results saved as a metadata column.

Examples

```
scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
scObj <- qs2::qs_read(scoPath)
gsPath <- system.file('extdata', 'geneSets.qs2', package='GSABenchmark')
geneSets <- qs2::qs_read(gsPath)
scObj <- runGSVA(scObj, geneSets)
```

`runJASMINE` *Run JASMINE*

Description

This function runs `JASMINE`.

Usage

```
runJASMINE(
  scObj,
  geneSets,
  slot = "data",
  method = c("oddsratio", "likelihood")
)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`geneSets`	A named list of gene sets.
`slot`	Gene expression slot.
`method`	One of ’oddsratio’ and ’likelihood’.

Details

Reimplemented from `https://github.com/NNoureen/JASMINE`. The method’s paper can be found at `https://doi.org/10.7554/eLife.71994`.

Value

A single-cell expression object with the results saved as a metadata column.

Examples

```
scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
scObj <- qs2::qs_read(scoPath)
gsPath <- system.file('extdata', 'geneSets.qs2', package='GSABenchmark')
geneSets <- qs2::qs_read(gsPath)
scObj <- runJASMINE(scObj, geneSets)
```

`runMDT` *Run MDT using decoupleR*

Description

This function runs `MDT` using `decoupleR`.

Usage

```
runMDT(scObj, geneSets, slot = "data", ...)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`geneSets`	A named list of gene sets.
`slot`	Gene expression slot.
`...`	Additional parameters passed to `runDecoupleRMethod`.

Value

A single-cell expression object with the results saved as a metadata column.

Examples

```
if (requireNamespace("ranger", quietly=TRUE)){
    scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
    scObj <- qs2::qs_read(scoPath)
    gsPath <- system.file('extdata', 'geneSets.qs2', package='GSABenchmark')
    geneSets <- qs2::qs_read(gsPath)
    scObj <- runMDT(scObj, geneSets)
}
```

`runMethodShuffle` *Run gene set analysis method on shuffled gene sets*

Description

This function runs a gene set analysis method on shuffled gene sets.

Usage

```
runMethodShuffle(
  scObj,
  labelCol,
  geneSets,
  gsaMethod,
  loss = c(0, 0.2),
  noise = c(0, 0.2),
  doGrid = TRUE,
  averageReplicates = TRUE,
  seeds = c(1, 2, 3),
  outputFun = identity
)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`labelCol`	The metadata column containing the ground truth annotation.
`geneSets`	A named list of gene sets.
`gsaMethod`	Name of the gene set analysis method.
`loss`	A numeric vector of gene loss values. Must be in [0, 1).
`noise`	A numeric vector of noise values. Must be in [0, 1).
`doGrid`	Whether to run the methods for each loss-noise combination.
`averageReplicates`	Whether to average replicates. If `TRUE`, a single set of scores will be returned for each loss-noise combination for which the method is run. If `FALSE`, distinct sets of scores will be returned for each replicate.
`seeds`	A numeric vector of random seeds passed to `hammers::shuffleGenes`. Its length determines the number of replicates.
`outputFun`	Choose between silently_run (suppress all warnings and messages) or identity (do not suppress them). Default is silently_run.

Value

A `Seurat` or `SingleCellExpression` object with the results of the runs stored as metadata columns.

Examples

```
scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
scObj <- qs2::qs_read(scoPath)
gsPath <- system.file('extdata', 'geneSets.qs2', package='GSABenchmark')
geneSets <- qs2::qs_read(gsPath)
scObj <- runMethodShuffle(scObj, 'label', geneSets, 'CSOA', 0.2, 0.2)
```

`runMLM` *Run MLM using decoupleR*

Description

This function runs `MLM` using `decoupleR`.

Usage

```
runMLM(scObj, geneSets, slot = "data", ...)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`geneSets`	A named list of gene sets.
`slot`	Gene expression slot.
`...`	Additional parameters passed to `runDecoupleRMethod`.

Value

A single-cell expression object with the results saved as a metadata column.

Examples

```
scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
scObj <- qs2::qs_read(scoPath)
gsPath <- system.file('extdata', 'geneSets.qs2', package='GSABenchmark')
geneSets <- qs2::qs_read(gsPath)
scObj <- runMLM(scObj, geneSets)
```

`runORA` *Run ORA using decoupleR*

Description

This function runs `ORA` using `decoupleR`.

Usage

```
runORA(scObj, geneSets, slot = "data", ...)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`geneSets`	A named list of gene sets.
`slot`	Gene expression slot.
`...`	Additional parameters passed to `runDecoupleRMethod`.

Value

A single-cell expression object with the results saved as a metadata column.

Examples

```
scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
scObj <- qs2::qs_read(scoPath)
gsPath <- system.file('extdata', 'geneSets.qs2', package='GSABenchmark')
geneSets <- qs2::qs_read(gsPath)
scObj <- runORA(scObj, geneSets)
```

`runPagoda2` *Run pagoda2*

Description

This function runs `pagoda2`.

Usage

```
runPagoda2(scObj, geneSets, slot = "data", ...)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`geneSets`	A named list of gene sets.
`slot`	Gene expression slot.
`...`	Additional arguments passed to `pagoda2::score.cells.puram`.

Value

A single-cell expression object with the results saved as a metadata column.

Examples

```
scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
scObj <- qs2::qs_read(scoPath)
gsPath <- system.file('extdata', 'geneSets.qs2', package='GSABenchmark')
geneSets <- qs2::qs_read(gsPath)
scObj <- runPagoda2(scObj, geneSets)
```

`runPLAGE` *Run PLAGE*

Description

This function runs `PLAGE` using `GSVA`.

Usage

```
runPLAGE(scObj, geneSets, slot = "data", ...)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`geneSets`	A named list of gene sets.
`slot`	Gene expression slot.
`...`	Additional parameters passed to `runGSVAMethod`.

Value

A single-cell expression object with the results saved as a metadata column.

Examples

```
scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
scObj <- qs2::qs_read(scoPath)
gsPath <- system.file('extdata', 'geneSets.qs2', package='GSABenchmark')
geneSets <- qs2::qs_read(gsPath)
scObj <- runPLAGE(scObj, geneSets)
```

`runSingscore` *Run Singscore*

Description

This function runs `Singscore`.

Usage

```
runSingscore(scObj, geneSets, slot = "data", ...)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`geneSets`	A named list of gene sets.
`slot`	Gene expression slot.
`...`	Additional arguments passed to `singscore::simpleScore`.

Value

A single-cell expression object with the results saved as a metadata column.

Examples

```
scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
scObj <- qs2::qs_read(scoPath)
gsPath <- system.file('extdata', 'geneSets.qs2', package='GSABenchmark')
geneSets <- qs2::qs_read(gsPath)
scObj <- runSingscore(scObj, geneSets)
```

`runSiPSiC` *Run SiPSiC*

Description

This function runs `SiPSiC`.

Usage

```
runSiPSiC(scObj, geneSets, slot = "counts", ...)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`geneSets`	A named list of gene sets.
`slot`	Gene expression slot.
`...`	Additional arguments passed to `SiPSiC::getPathwayScores`.

Value

A single-cell expression object with the results saved as a metadata column.

Examples

```
scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
scObj <- qs2::qs_read(scoPath)
gsPath <- system.file('extdata', 'geneSets.qs2', package='GSABenchmark')
geneSets <- qs2::qs_read(gsPath)
scObj <- runSiPSiC(scObj, geneSets)
```

`runssGSEA` *Run ssGSEA*

Description

This function runs `ssGSEA` using `escape`.

Usage

```
runssGSEA(scObj, geneSets, slot = "data", ...)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`geneSets`	A named list of gene sets.
`slot`	Gene expression slot.
`...`	Additional parameters passed to `runEscapeMethod`.

Value

A single-cell expression object with the results saved as a metadata column.

Examples

```
scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
scObj <- qs2::qs_read(scoPath)
gsPath <- system.file('extdata', 'geneSets.qs2', package='GSABenchmark')
geneSets <- qs2::qs_read(gsPath)
scObj <- runssGSEA(scObj, geneSets)
```

`runUCell` *Run UCell*

Description

This function runs `UCell` using `escape`.

Usage

```
runUCell(scObj, geneSets, slot = "data", ...)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`geneSets`	A named list of gene sets.
`slot`	Gene expression slot.
`...`	Additional parameters passed to `runEscapeMethod`.

Value

A single-cell expression object with the results saved as a metadata column.

Examples

```
if (requireNamespace("UCell", quietly=TRUE)){
    scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
    scObj <- qs2::qs_read(scoPath)
    gsPath <- system.file('extdata', 'geneSets.qs2', package='GSABenchmark')
    geneSets <- qs2::qs_read(gsPath)
    scObj <- runUCell(scObj, geneSets)
    }
```

`runUDT` *Run UDT using decoupleR*

Description

This function runs `UDT` using `decoupleR`.

Usage

```
runUDT(scObj, geneSets, slot = "data", ...)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`geneSets`	A named list of gene sets.
`slot`	Gene expression slot.
`...`	Additional parameters passed to `runDecoupleRMethod`.

Value

A single-cell expression object with the results saved as a metadata column.

Examples

```
if (requireNamespace("rpart", quietly=TRUE)){
    scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
    scObj <- qs2::qs_read(scoPath)
    gsPath <- system.file('extdata', 'geneSets.qs2', package='GSABenchmark')
    geneSets <- qs2::qs_read(gsPath)
    scObj <- runUDT(scObj, geneSets)
    }
```

`runVAM` *Run VAM*

Description

This function runs `VAM`.

Usage

```
runVAM(scObj, geneSets, slot = "data", ...)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`geneSets`	A named list of gene sets.
`slot`	Gene expression slot.
`...`	Additional arguments passed to `VAM::vam`.

Value

A single-cell expression object with the results saved as a metadata column.

Examples

```
scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
scObj <- qs2::qs_read(scoPath)
gsPath <- system.file('extdata', 'geneSets.qs2', package='GSABenchmark')
geneSets <- qs2::qs_read(gsPath)
scObj <- runVAM(scObj, geneSets)
```

`runZscore` *Run Zscore*

Description

This function runs `Zscore` using `GSVA`.

Usage

```
runZscore(scObj, geneSets, slot = "data", ...)
```

Arguments

`scObj`	A `Seurat` or `SingleCellExperiment` object.
`geneSets`	A named list of gene sets.
`slot`	Gene expression slot.
`...`	Additional parameters passed to `runGSVAMethod`.

Value

A single-cell expression object with the results saved as a metadata column.

Examples

```
scoPath <- system.file('extdata', 'scObj.qs2', package='GSABenchmark')
scObj <- qs2::qs_read(scoPath)
gsPath <- system.file('extdata', 'geneSets.qs2', package='GSABenchmark')
geneSets <- qs2::qs_read(gsPath)
scObj <- runZscore(scObj, geneSets)
```

`scorePlot` *Plot a data frame consisting of gene set analysis method scores This function plots a data frame consisting of method scores with methods as rows, gene sets and the gene set average as columns.*

Description

Plot a data frame consisting of gene set analysis method scores This function plots a data frame consisting of method scores with methods as rows, gene sets and the gene set average as columns.

Usage

```
scorePlot(
  scoreDF,
  title = NULL,
  xLab = "Score",
  yLab = "Method",
  isDecreasing = FALSE,
  palette = "grDevices::Plasma",
  legendLab = "Gene set",
  pointSize = 1.5
)
```

Arguments

`scoreDF`	A summary data frame.
`title`	Plot title.
`xLab`	x axis label.
`yLab`	y axis label.
`isDecreasing`	Logical; whether the methods should be displayed on the plot in decreasing order of the obtained average scores. If `FALSE` (as default), the methods will be displayed in increasing order of the average scores.
`palette`	Color palette.
`legendLab`	Legend label.
`pointSize`	Point size.

Value

A ggplot object.

Examples

```
sPath <- system.file('extdata', 'smr.qs2', package='GSABenchmark')
smr <- qs2::qs_read(sPath)
scorePlot(smr[[1]][[1]])
```

`supportedMethods` *Show supported methods*

Description

This function shows the methods currently supported by GSABenchmark.

Usage

```
supportedMethods()
```

Value

A character vector of supported methods.

Examples

```
supportedMethods()
```

`timePlot` *Plot a data frame consisting of gene set analysis method running times This function plots data frame consisting of method running times with methods as rows, gene sets and the gene set average as columns.*

Description

Plot a data frame consisting of gene set analysis method running times This function plots data frame consisting of method running times with methods as rows, gene sets and the gene set average as columns.

Usage

```
timePlot(efBenchmark, title = NULL, xLab = "Running time (s)", ...)
```

Arguments

`efBenchmark`	A list of dataframes generated with `efficiencyBenchmark`.
`title`	Plot title.
`xLab`	x axis label.
`...`	Additional parameters to be passed to `scorePlot`.

Value

A ggplot object.

Examples

```
sPath <- system.file('extdata', 'smr.qs2', package='GSABenchmark')
smr <- qs2::qs_read(sPath)
timePlot(smr[[5]])
```

Index