Package: recall (via r-universe)

May 28, 2026

Title Calibrated Clustering with Artificial Variables to Avoid Over-Clustering in Single-Cell RNA-Sequencing

Version 0.1.0

Description recall (Calibrated Clustering with Artificial Variables) is a method for protecting against over-clustering by controlling for the impact of double-dipping. The approach can be applied to any clustering algorithm (implemented are the Louvain and Leiden algorithms with plans for K-means, and hierarchical clustering algorithms). The method provides state-of-the-art clustering performance and can rapidly analyze large-scale scRNA-seq studies and is compatible with the Seurat library (V4 and V5).

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports Matrix, Seurat (>= 4.0.0), SingleCellExperiment, scDesign3, SummarizedExperiment, MASS, fitdistrplus, lamW, knockoff, future, stats, cli, stringr, countsplit, utils

License MIT + file LICENSE

Suggests knitr, markdown

Remotes scDesign3=github::SONGDONGYUAN1994/scDesign3

VignetteBuilder knitr

URL `https://zaoqu-liu.github.io/recall/`, `https://github.com/Zaoqu-Liu/recall`

BugReports `https://github.com/Zaoqu-Liu/recall/issues`

Config/pak/sysreqs cmake libglpk-dev make libicu-dev libpng-dev libuv1-dev libxml2-dev libssl-dev python3 zlib1g-dev

Repository https://zaoqu-liu.r-universe.dev

Date/Publication 2026-01-23 18:03:47 UTC

RemoteUrl https://github.com/Zaoqu-Liu/recall

RemoteRef main

RemoteSha ebad0cdf9754bb32a3a756d5af31e7df8fad97c3

Contents

`compute_knockoff_filter` *Returns the genes selected by the knockoff filter*

Description

Given two Seurat objects, returns the the genes selected by the knockoff filter and their W statistics.

Usage

```
compute_knockoff_filter(
  seurat_obj,
  cluster1,
  cluster2,
  q,
  return_all = FALSE,
  num_cores = 1,
  shared_memory_max
)
```

Arguments

`seurat_obj`	A Seurat object
`cluster1`	The Idents of the cluster of interest in seurat_obj1
`cluster2`	The Idents of the cluster of interest in seurat_obj2
`q`	The desired rate to control the FDR at
`return_all`	Determines if the returned object will contain all genes or just the selected genes.
`num_cores`	The number of cores for computing marker genes in parallel.
`shared_memory_max`	The maximum size for shared global variables.

Value

todo

`estimate_negative_binomial`

Maximum likelihood estimation for the negative binomial distribution.

Description

Given data, computes the maximum likelihood estimators for the negative binomial distribution with parameters: size and mu.

Usage

```
estimate_negative_binomial(data, verbose = FALSE)
```

Arguments

`data`	The data to estimate parameters from.
`verbose`	Whether or not to show all logging.

Value

Maximum likelihood estimators size and mu for the negative binomial distribution

---

`estimate_negative_binomial_copula`

todo

Description

Given data, computes todo

Given data, computes todo

Given data, computes todo

Given data, computes todo

Usage

```
estimate_zi_poisson_copula(data_matrix, cores)

estimate_negative_binomial_copula(data_matrix, cores)

estimate_poisson_copula(data_matrix, cores)

estimate_gaussian_copula(data_matrix, cores)
```

Arguments

`data_matrix`	The data to estimate parameters from.
`cores`	The number of CPU cores to use in estimation by scDesign3.

Value

todo

todo

todo

todo

`estimate_zi_poisson` *Maximum likelihood estimation for the zero-inflated Poisson distribution with Poisson parameter lambda and zero proportion prop.zero.*

Description

Given data, computes the maximum likelihood estimators for the zero-inflated Poisson distribution.

Usage

```
estimate_zi_poisson(data)
```

Arguments

`data`	The data to estimate parameters from.

Value

Maximum likelihood estimators of the zero-inflated Poisson distribution

---

`FindClustersCountsplit` *Runs a typical Seurat workflow on a Seurat object (up to dimensionality reduction and clustering).*

Description

Given a Seurat object, returns a new Seurat that has been normalized, had variable features identified, scaled, had principal components computed, hadclusters identified, and had tSNE and UMAP embeddings determined.

Usage

```
FindClustersCountsplit(
  seurat_obj,
  resolution_start = 0.8,
  reduction_percentage = 0.2,
  num_clusters_start = 20,
  dims = 1:10,
  algorithm = "louvain",
  null_method = "ZIP",
  assay = "RNA",
  cores = 1,
  shared_memory_max = 8000 * 1024^2,
  verbose = TRUE
)
```

Arguments

`seurat_obj`	The Seurat object that will be analyzed.
`resolution_start`	The starting resolution to be used for the clustering algorithm (Louvain and Leiden algorithms).
`reduction_percentage`	The amount that the starting parameter will be reduced by after each iteration (between 0 and 1).
`num_clusters_start`	The starting number of clusters to be used for the clustering algorithm (K-means and Hierarchical clustering algorithms).
`dims`	The dimensions to use as input features (i.e. 1:10).
`algorithm`	The clustering algorithm to be used.
`null_method`	The generating distribution for the synthetic null variables (ZIP, NB, ZIP-copula, NB-copula)
`assay`	The assay to generate artificial variables from.
`cores`	The number of cores to compute marker genes in parallel.
`shared_memory_max`	The maximum size for shared global variables. Increased this variable if you see the following error: The total size of the X globals that need to be exported for the future expression ('FUN()') is X GiB. This exceeds the maximum allowed size of 500.00 MiB (option 'future.globals.maxSize'). The X largest globals are ...
`verbose`	Whether or not to show all logging.

Value

Returns a Seurat object where the idents have been updated with the clusters determined via the countsplit algorithm. Latest clustering results will be stored in the object metadata under countsplit_clusters'. Note that 'countsplit_clusters' will be overwritten ever time FindClustersCountsplit is run.

`FindClustersRecall` *Runs a typical Seurat workflow on a Seurat object (up to dimensionality reduction and clustering).*

Description

Given a Seurat object, returns a new Seurat that has been normalized, had variable features identified, scaled, had principal components computed, hadclusters identified, and had tSNE and UMAP embeddings determined.

Usage

```
FindClustersRecall(
  seurat_obj,
  resolution_start = 0.8,
  reduction_percentage = 0.2,
  num_clusters_start = 20,
  dims = 1:10,
  algorithm = "louvain",
  null_method = "ZIP",
  assay = "RNA",
  cores = 1,
  shared_memory_max = 8000 * 1024^2,
  verbose = TRUE
)
```

Arguments

`seurat_obj`	The Seurat object that will be analyzed.
`resolution_start`	The starting resolution to be used for the clustering algorithm (Louvain and Leiden algorithms).
`reduction_percentage`	The amount that the starting parameter will be reduced by after each iteration (between 0 and 1).
`num_clusters_start`	The starting number of clusters to be used for the clustering algorithm (K-means and Hierarchical clustering algorithms).
`dims`	The dimensions to use as input features (i.e. 1:10).
`algorithm`	The clustering algorithm to be used.
`null_method`	The generating distribution for the synthetic null variables (ZIP, NB, ZIP-copula, NB-copula)
`assay`	The assay to generate artificial variables from.
`cores`	The number of cores to compute marker genes in parallel.

`shared_memory_max`	The maximum size for shared global variables. Increased this variable if you see the following error: The total size of the X globals that need to be exported for the future expression ('FUN()') is X GiB. This exceeds the maximum allowed size of 500.00 MiB (option 'future.globals.maxSize'). The X largest globals are ...
`verbose`	Whether or not to show all logging.

Value

Returns a Seurat object where the idents have been updated with the clusters determined via the recall algorithm. Latest clustering results will be stored in the object metadata under recall_clusters'. Note that 'recall_clusters' will be overwritten ever time FindClustersRecall is run.

`get_seurat_obj_with_artificial_variables`

Returns a Seurat object that contains additional (fake) RNA expression counts.

Description

Given a Seurat object, returns a new Seurat object whose RNA expression counts includes the variable features from the original object and an equal number of artificial features.

Usage

```
get_seurat_obj_with_artificial_variables(
  seurat_obj,
  assay = "RNA",
  null_method = "ZIP",
  verbose = TRUE,
  cores
)
```

Arguments

`seurat_obj`	A Seurat object containing RNA expression counts.
`assay`	The assay to generate artificial variables from.
`null_method`	The generating distribution for the synthetic null variables (ZIP, NB, ZIP-copula, NB-copula)
`verbose`	Whether or not to show logging.
`cores`	The number of cores to use in generating synthetic null variables.

Value

A Seurat object that contains the original variable features and an equal number of artificial features.

`rzipoisson` *Random data generation for the zero-infalted Poisson distribution with Poisson parameter lambda and zero proportion prop.zero.*

Description

Given the number of samples desired, a Poisson parameter, lambda, and a zero proportion, prop.zero, simulates the number of desired samples from ZIP(lambda, prop.zero).

Usage

```
rzipoisson(n, lambda, prop.zero)
```

Arguments

`n`	The number of samples to be simulated.
`lambda`	The Poisson rate parameter.
`prop.zero`	The proportion of excess zeroes.

Value

Simulated data from ZIP(lambda, prop.zero).

`seurat_workflow` *Runs a typical Seurat workflow on a Seurat object (up to dimensionality reduction and clustering).*

Description

Given a Seurat object, returns a new Seurat that has been normalized, had variable features identified, scaled, had principal components computed, had clusters identified, and had tSNE and UMAP embeddings determined.

Usage

```
seurat_workflow(
  seurat_obj,
  num_variable_features,
  resolution_param = 0.8,
  visualization_method = "umap",
  num_dims = 10,
  algorithm = "louvain"
)
```

Arguments

`seurat_obj`	A Seurat object that will be analyzed.
`num_variable_features`	The number of variable features to use in the analysis.
`resolution_param`	The resolution parameter to use when clustering.
`visualization_method`	Either "umap" or "tsne".
`num_dims`	The number of principal components to use.
`algorithm`	The clustering algorithm to use, either "louvain" or "leiden".

Value

A Seurat object containing the relevant analysis results.

Index