Package: scPAS (via r-universe)

May 23, 2026

Type Package

Title Single-Cell Phenotype-Associated Subpopulation Identifier

Version 1.0.4

Maintainer Zaoqu Liu <liuzaoqu@163.com>

Description Identifies phenotype-associated cell subpopulations from single-cell RNA-seq data by integrating bulk RNA-seq data with phenotype information. This package uses network-regularized sparse regression to quantify the strength of association between each cell and a phenotype (e.g., disease stage, tumor metastasis, treatment response, survival outcomes). Compatible with both Seurat v4 (4.0.0+) and Seurat v5 (5.0.0+). The method supports Gaussian (continuous), binomial (binary), and Cox (survival) regression families. Full cross-platform compatibility (Windows, macOS, Linux).

URL https://github.com/Zaoqu-Liu/scPAS

BugReports https://github.com/Zaoqu-Liu/scPAS/issues

Depends R (>= 4.0.0)

Imports Rcpp (>= 1.0.0), Matrix, methods, stats, utils, Seurat (>= 4.0.0), SeuratObject (>= 4.0.0), preprocessCore, survival

Suggests ALRA, future, future.apply, testthat (>= 3.0.0), knitr, rmarkdown, ggplot2, patchwork, RColorBrewer, survminer, dplyr, reshape2

VignetteBuilder knitr

LinkingTo Rcpp, RcppEigen

License GPL-3

Encoding UTF-8

LazyData true

NeedsCompilation yes

RoxygenNote 7.3.3

SystemRequirements GNU make

Biarch true

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 10:16:15 UTC

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

RemoteRef HEAD

RemoteSha 8f4f190d3e9b11b4b9f4a128fc8a0b66c1b22e75

Contents

`imputation` *The function of imputaion.*

Description

The function of imputaion.

Usage

```
imputation(obj, assay = "RNA", method = c("KNN", "ALRA"))
```

Arguments

`obj`	A seurat object.
`assay`	The assay for imputation. The default is 'RNA'.
`method`	The method for imputation. The default is 'RNA'.

Value

A seurat object after imputaion.

`imputation_ALRA` *A method for imputation of missing values in single cell RNA-sequencing data based on ALRA.*

Description

A method for imputation of missing values in single cell RNA-sequencing data based on ALRA.

Usage

```
imputation_ALRA(obj, assay = "RNA")
```

Arguments

`obj`	A seurat object.
`assay`	The assay for imputation. The default is 'RNA'.

Value

A seurat object after imputaion.

`imputation_KNN` *A method for imputation of missing values in single cell RNA-sequencing data based on the average expression value of nearest neighbor cells.*

Description

A method for imputation of missing values in single cell RNA-sequencing data based on the average expression value of nearest neighbor cells.

Usage

```
imputation_KNN(obj, assay = "RNA", LogNormalized = TRUE)
```

Arguments

`obj`	A seurat object.
`assay`	The assay for imputation. The default is 'RNA'.
`LogNormalized`	Whether the data is LogNormalized.

Value

A seurat object after imputaion.

`run_Seurat` *Preprocess the single-cell raw data using functions in the* `Seurat` *package*

Description

This function provide a simplified-version of Seurat analysis pipeline for single-cell RNA-seq data. It contains the following steps in the pipeline:

- Create a `Seurat` object from raw data.
- Normalize the count data present in a given assay.
- Identify the variable features.
- Scales and centers features in the dataset.
- Run a PCA dimensionality reduction.
- Constructs a Shared Nearest Neighbor (SNN) Graph for a given dataset.
- Identify clusters of cells by a shared nearest neighbor (SNN) modularity optimization based clustering algorithm.
- Run t-distributed Stochastic Neighbor Embedding (t-SNE) dimensionality reduction on selected features.
- Runs the Uniform Manifold Approximation and Projection (UMAP) dimensional reduction technique.

Usage

```
run_Seurat(
  counts,
  project = "Single_Cell",
  min.cells = 400,
  min.features = 200,
  meta.data = NULL,
  normalization.method = "LogNormalize",
  scale.factor = 10000,
  selection.method = "vst",
  resolution = 0.6,
  dims_Neighbors = 1:10,
  dims_TSNE = 1:10,
  dims_UMAP = 1:10,
  verbose = TRUE
)
```

Arguments

`counts`	A `matrix`-like object with unnormalized data with cells as columns and features as rows.
`project`	Project name for the `Seurat` object.

`min.cells` Include features detected in at least this many cells. Will subset the counts matrix as well. To reintroduce excluded features, create a new object with a lower cutoff.

`min.features` Include cells where at least this many features are detected.

`meta.data` meta data of single cell data.

`normalization.method`
Method for normalization.

- LogNormalize: Feature counts for each cell are divided by the total counts for that cell and multiplied by the scale.factor. This is then natural-log transformed using log1p.
- CLR: Applies a centered log ratio transformation.
- RC: Relative counts. Feature counts for each cell are divided by the total counts for that cell and multiplied by the scale.factor. No log-transformation is applied. For counts per million (CPM) set `scale.factor = 1e6`.

`scale.factor` Sets the scale factor for cell-level normalization.

`selection.method`
How to choose top variable features. Choose one of :

- vst: First, fits a line to the relationship of log(variance) and log(mean) using local polynomial regression (loess). Then standardizes the feature values using the observed mean and expected variance (given by the fitted line). Feature variance is then calculated on the standardized values after clipping to a maximum (see clip.max parameter).
- mean.var.plot (mvp): First, uses a function to calculate average expression (mean.function) and dispersion (dispersion.function) for each feature. Next, divides features into num.bin (deafult 20) bins based on their average expression, and calculates z-scores for dispersion within each bin. The purpose of this is to identify variable features while controlling for the strong relationship between variability and average expression.
- dispersion (disp): selects the genes with the highest dispersion values

`resolution` Value of the resolution parameter, use a value above (below) 1.0 if you want to obtain a larger (smaller) number of communities.

`dims_Neighbors` Dimensions of reduction to use as input.

`dims_TSNE` Which dimensions to use as input features for t-SNE.

`dims_UMAP` Which dimensions to use as input features for UMAP.

`verbose` Print output.

Value

A `Seurat` object containing cell-cell similarity network, t-SNE and UMAP representations.

scPAS *scPAS : A tool for identifying Phenotype-Associated cell Subpopulations from single-cell sequencing data by integrating bulk data*

Description

scPAS : A tool for identifying Phenotype-Associated cell Subpopulations from single-cell sequencing data by integrating bulk data

Usage

```
scPAS(
  bulk_dataset,
  sc_dataset,
  phenotype,
  assay = "RNA",
  tag = NULL,
  nfeature = NULL,
  do_imputation = TRUE,
  imputation_method = c("KNN", "ALRA"),
  alpha = NULL,
  network_class = c("SC", "bulk"),
  independent = TRUE,
  family = c("gaussian", "binomial", "cox"),
  permutation_times = 2000,
  FDR.threshold = 0.05,
  n_cores = 1
)
```

Arguments

bulk_dataset	Matrix. Bulk expression matrix of related disease. Each row represents a gene and each column represents a sample. The input expression values are continuous, such as microarray fluorescent units in logarithmic scale, RNA-seq log-CPMs, log-RPKMs or log-TPMs.
sc_dataset	Matrix or seurat object. Single-cell RNA-seq expression matrix of related disease. Each row represents a gene and each column represents a sample. A Seurat object that contains the preprocessed data and constructed network is preferred. Otherwise, a cell-cell similarity network is constructed based on the input matrix.Otherwise, the raw count expression matrix will be processed by using Seurat's default parameters. See run_Seurat for details.
phenotype	Phenotype annotation of each bulk sample. It can be a continuous dependent variable, binary group indicator vector, or clinical survival data: • Continuous dependent variable. Should be a quantitative vector for `family = gaussian`.

- Binary group indicator vector. Should be either a 0-1 encoded vector or a factor with two levels for `family = binomial`.
- Clinical survival data. Should be a two-column matrix with columns named 'time' and 'status'. The latter is a binary variable, with '1' indicating event (e.g.recurrence of cancer or death), and '0' indicating right censored. The function `Surv()` in package survival produces such a matrix.

`assay` Name of Assay to get.

`tag` Names for each phenotypic group. Used for logistic regressions only.

`nfeature` Numeric. The Number of features to select as top variable features in sc_dataset. Top variable features will be used to intersect with the features of bulk_dataset. Default is NULL.All features will be used.

`do_imputation` Logical. Whether to perform imputation on single-cell data (default: TRUE).

`imputation_method` Character. Name of alternative method for imputation.

`alpha` Numeric. Parameter used to balance the effect of the l1 norm and the network-based penalties. It can be a number or a searching vector. If `alpha = NULL`, a default searching vector is used. The range of alpha is in `[0,1]`. A larger alpha lays more emphasis on the l1 norm.

`network_class` The source of feature-feature similarity network. By default this is set to `sc` and the other one is `bulk`.

`independent` Logical. The background distribution of risk scores is constructed independently of each cell.

`family` Character. Response type for the regression model. It depends on the type of the given phenotype and can be `family = gaussian` for linear regression, `family = binomial` for classification, or `family = cox` for Cox regression.

`permutation_times` Integer. Number of permutation iterations for statistical significance testing (default: 2000). Higher values increase accuracy but also computation time. Recommended: 1000-5000. For faster testing, use 500-1000.

`FDR.threshold` Numeric. FDR value threshold for identifying phenotype-associated cells. The default is 0.05.

`n_cores` Integer. Number of CPU cores to use for parallel permutation test (default: 1 for sequential processing). Setting n_cores > 1 enables parallel computing which can significantly speed up the analysis (2-4x faster with 4 cores). Requires 'future' and 'future.apply' packages.

Value

This function returns a Seurat object with the following components added to :

`scPAS_para` A list contains the final model parameters added to misc.

`PAS result` A data frame containing risk scores (scPAS_RS), normalized risk scores (scPAS_NRS), p-value (scPAS_Pvalue) , adjusted p-value (scPAS_FDR) cell classification labels (scPAS) added to metaData.

scPAS.prediction *scPAS.prediction: A function that uses the scPAS model to make predictions on independent data*

Description

scPAS.prediction: A function that uses the scPAS model to make predictions on independent data

Usage

```
scPAS.prediction(
  model,
  test.data,
  assay = "RNA",
  FDR.threshold = 0.05,
  do_imputation = FALSE,
  imputation_method = "KNN",
  independent = TRUE,
  permutation_times = 2000,
  n_cores = 1
)
```

Arguments

model	Seurat object. A Seurat object containing the scPAS model (from running scPAS()).
test.data	Matrix or Seurat object. Single-cell RNA-seq expression matrix of related disease. Each row represents a gene and each column represents a sample. A Seurat object that contains the preprocessed data and constructed network is preferred.
assay	Name of Assay to get.
FDR.threshold	Numeric. FDR value threshold for identifying phenotype-associated cells. The default is 0.05.
do_imputation	Logical. Whether to perform imputation on the test data (default: FALSE).
imputation_method	Character. Imputation method: "KNN" or "ALRA".
independent	Logical. Whether to compute background distribution independently for each cell.
permutation_times	Integer. Number of permutations for significance testing (default: 2000).
n_cores	Integer. Number of CPU cores for parallel processing (default: 1).

Value

A seurat object or data frame containing the forecast results.

`sparse.cor` *A function compute the correlation of a sparse matrix.*

Description

A function compute the correlation of a sparse matrix.

Usage

```
sparse.cor(x)
```

Arguments

`x`	Matrix. Normalized single cell expression profile extracted from Seurat object.

Value

A correlation matrix.

Index