Package 'waddR'

November 12, 2026

Type Package

Title Statistical tests for detecting differential distributions based on the 2-Wasserstein distance

Version 1.26.0

Description The package offers statistical tests based on the 2-Wasserstein distance for detecting and characterizing differences between two distributions given in the form of samples. Functions for calculating the 2-Wasserstein distance and testing for differential distributions are provided, as well as a specifically tailored test for differential expression in single-cell RNA sequencing data.

License MIT + file LICENSE

biocViews Software, StatisticalMethod, SingleCell, DifferentialExpression

BugReports https://github.com/goncalves-lab/waddR/issues

URL https://github.com/goncalves-lab/waddR.git

Encoding UTF-8

Additional_repositories https://www.bioconductor.org/

Imports Rcpp (>= 1.0.1), arm (>= 1.10-1), eva, BiocFileCache (>= 2.6.0), BiocParallel, SingleCellExperiment, parallel, methods, stats

Depends R (>= 3.6.0)

Remotes url::https://cran.r-project.org/src/contrib/Archive/eva/eva_0.2.5.tar.gz

Suggests knitr, devtools, testthat, roxygen2, rprojroot, rmarkdown, scater

LinkingTo Rcpp, RcppArmadillo,

VignetteBuilder knitr

RoxygenNote 7.2.3

git_url https://git.bioconductor.org/packages/waddR

git_branch RELEASE_3_23

git_last_commit 5f7e983

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Roman Schefzik [aut],
Julian Flesch [cre]

Maintainer Julian Flesch <julianflesch@gmail.com>

Contents

waddR-package . . . 2
.brownianBridgeEmpcdf . . . 4
.combinePVal . . . 5
.fishersCombinedPval . . . 5
.gpdFittedPValue . . . 6
.quantileCorrelation . . . 7
.relativeError . . . 7
.testWass . . . 8
.wassersteinTestAsy . . . 9
.wassersteinTestSp . . . 10
.wassPermProcedure . . . 12
brownianBridgeEmpcdf.url . . . 12
permutations . . . 13
squared_wass_approx . . . 13
squared_wass_decomp . . . 14
testZeroes . . . 15
wasserstein.sc . . . 17
wasserstein.test . . . 21
wasserstein_metric . . . 24

Index **26**

`waddR-package` *waddR: Statistical tests for detecting differential distributions based on the 2-Wasserstein distance*

Description

The package offers statistical tests based on the 2-Wasserstein distance for detecting and characterizing differences between two distributions given in the form of samples. Functions for calculating the 2-Wasserstein distance and testing for differential distributions are provided, as well as a specifically tailored test for differential expression in single-cell RNA sequencing data.

Details

The waddR package provides tools to address the following tasks:

1. Computation of the 2-Wasserstein distance
2. Two-sample tests to check for differences between two distributions
3. Detection of differential gene expression distributions in single-cell RNA sequencing data

1. 2-Wasserstein distance functions

The 2-Wasserstein distance is a metric to quantify the difference between two distributions, representing e.g. two different conditions A and B. The waddR package specifically considers the squared 2-Wasserstein distance which can be decomposed into location, size, and shape terms, thus providing a characterization of potential differences. It offers three functions to calculate the (squared) 2-Wasserstein distance, which are implemented in C++ and exported to R with Rcpp for faster computation. `wasserstein_metric` is a C++ reimplementation of the `wasserstein1d` function from the R package `transport`. The functions `squared_wass_approx` and `squared_wass_decomp` compute approximations of the squared 2-Wasserstein distance, with `squared_wass_decomp` also returning the decomposition terms for location, size, and shape.

See `?wasserstein_metric`, `?squared_wass_aprox`, and `?squared_wass_decomp` as well as the accompanying paper Schefzik et al. (2020).

2. Testing for differences between two distributions

The waddR package provides two testing procedures using the 2-Wasserstein distance to test whether two distributions F_A and F_B given in the form of samples are different by testing the null hypothesis $H_0 : F_A = F_B$ against the alternative hypothesis $H_1 : F_A \neq F_B$.

The first, semi-parametric (SP), procedure uses a permutation-based test combined with a generalized Pareto distribution approximation to estimate small p-values accurately.

The second procedure uses a test based on asymptotic theory (ASY) which is valid only if the samples can be assumed to come from continuous distributions.

See `?wasserstein.test` for more details.

3. Testing for differences between two distributions in the context of single-cell RNA sequencing (scRNA-seq) data

The waddR package provides an adaptation of the semi-parametric testing procedure based on the 2-Wasserstein distance which is specifically tailored to identify differential distributions in scRNA-seq data. In particular, a two-stage (TS) approach is implemented that takes account of the specific nature of scRNA-seq data by separately testing for differential proportions of zero gene expression (using a logistic regression model) and differences in non-zero gene expression (using the semi-parametric 2-Wasserstein distance-based test) between two conditions.

See `?wasserstein.sc` and `?testZeroes` for more details.

References

Schefzik, R., Flesch, J., and Goncalves, A. (2020). waddR: Using the 2-Wasserstein distance to identify differences between distributions in two-sample testing, with application to single-cell RNA-sequencing data.

Author(s)

Maintainer: Julian Flesch <julianflesch@gmail.com>

Authors:

- Roman Schefzik <roman.schefzik@medma.uni-heidelberg.de>

See Also

Useful links:

- https://github.com/goncalves-lab/waddR.git
- Report bugs at https://github.com/goncalves-lab/waddR/issues

---

`.brownianBridgeEmpcdf` *Compute value of the asymptotic CDF occuring in the asymptotic theory-based test*

Description

Computes the values of the cumulative distribution function (CDF) of the integral over the squared standard Brownian bridge in the unit interval, where the computation is based on Monte Carlo simulations. This CDF occurs as an asymptotic distribution in the asymptotic theory-based test using the 2-Wasserstein distance, see Schefzik et al. (2020) for details. It is used to determine the corresponding p-values in the function `.wassersteinTestAsy`.

Usage

```
.brownianBridgeEmpcdf(v)
```

Arguments

v	a number

Value

Value at v of the asymptotic CDF

References

Schefzik, R., Flesch, J., and Goncalves, A. (2020). waddR: Using the 2-Wasserstein distance to identify differences between distributions in two-sample testing, with application to single-cell RNA-sequencing data.

.combinePVal *Compute combined p-value using Fisher's method for several pairs of p-values*

Description

For a given set of N pairs of p-values, aggregates each respective pair of p-values into a combined p-value according to Fisher's method

Usage

```
.combinePVal(r, s)
```

Arguments

r	vector of length N of the p-values corresponding to the first test
s	vector of length N of the p-values corresponding to the second test

Details

For a given set of N pairs of p-values, aggregates each respective pair of p-values into a combined p-value according to Fisher's method. Applies the function `.fishersCombinedPval` to a whole set of N pairs of p-values.

Value

A vector of length N of the combined p-values

.fishersCombinedPval *Compute combined p-value using Fisher's method*

Description

Aggregates two p-values into a combined p-value according to Fisher's method

Usage

```
.fishersCombinedPval(x)
```

Arguments

x	vector of the two p-values that are to be aggregated

Details

Aggregates two p-values into a combined p-value according to Fisher's method

Value

The combined p-value

`.gpdFittedPValue` *Compute p-value based on generalized Pareto distribution fitting*

Description

Computes a p-value based on a generalized Pareto distribution (GPD) fitting. This procedure may be used in the semi-parametric 2-Wasserstein distance-based test to estimate small p-values accurately, instead of obtaining the p-value from a permutation test.

Usage

```
.gpdFittedPValue(val, distr.ordered)
```

Arguments

`val`	value of a specific test statistic, based on original group labels
`distr.ordered`	vector of values, in decreasing order, of the test statistic obtained by repeatedly permuting the original group labels

Value

A vector of three, see Schefzik et al. (2020) for details:

- pvalue.gpd: p-value obtained when using the GPD fitting test
- ad.pval: p-value of the Anderson-Darling test to check whether the GPD actually fits the data well
- N.exc: number of exceedances (starting with 250 and iteratively decreased by 10 if necessary) that are required to obtain a good GPD fit, i.e. p-value of Anderson-Darling test ≥ 0.05

References

Schefzik, R., Flesch, J., and Goncalves, A. (2020). waddR: Using the 2-Wasserstein distance to identify differences between distributions in two-sample testing, with application to single-cell RNA-sequencing data.

.quantileCorrelation *Compute the quantile-quantile correlation*

Description

Computes the quantile-quantile correlation of two samples x and y, using the quantile type 1 implementation in R and the Pearson correlation.

Usage

```
.quantileCorrelation(x, y, pr = NULL)
```

Arguments

x	numeric vector
y	numeric vector
pr	levels at which the quantiles of x and y are computed; by default, 1000 equidistant quantiles at levels $\frac{k-0.5}{1000}$, where $k = 1, \ldots, 1000$, are used

Value

quantile-quantile correlation of x and y

.relativeError *Compute relative error*

Description

Computes the relative error between two numbers x and y.

Usage

```
.relativeError(x, y)
```

Arguments

x	a number
y	a number

Details

The relative error e is defined as $e = |1 - x/y|$

Value

The relative error between x and y

`.testWass` *Check for differential distributions in single-cell RNA sequencing data via a semi-paramteric test using the 2-Wasserstein distance*

Description

Two-sample test for single-cell RNA-sequencing data to check for differences between two distributions using the 2-Wasserstein distance: Semi-parametric implementation using a permutation test with a generalized Pareto distribution (GPD) approximation to estimate small p-values accurately

Usage

```
.testWass(dat, condition, permnum, inclZero = TRUE, seed = NULL)
```

Arguments

`dat`	matrix of single-cell RNA-sequencing expression data, with rowas corresponding to genes and columns corresponding to cells (samples)
`condition`	vector of condition labels
`permnum`	number of permutations used in the permutation testing procedure
`inclZero`	logical; if TRUE, a one-stage method is performed, i.e. the semi-parametric test based on the 2-Wasserstein distance is applied to all (zero and non-zero) expression values; if FALSE, a two-stage method is performed, i.e. the semi-parametric test based on the 2-Wasserstein distance is applied to non-zero expression values only, and a separate test for differential proportions of zero expression using logistic regression is conducted; default is TRUE
`seed`	Number to be used as a L'Ecuyer-CMRG seed, which itself seeds the generation of an nextRNGStream() for each gene. Internally, when this argument is given, a seed is specified by calling 'RNGkind("L'Ecuyer-CMRG")' followed by 'set.seed(seed)'. The 'RNGkind' and '.Random.seed' will be reset on termination of this function. Default is NULL, and no seed is set.

Details

Details concerning the testing procedure for single-cell RNA-sequencing data can be found in Schefzik et al. (2021) and in the description of the details of the function `wasserstein.sc`.

Value

Matrix, where each row contains the testing results of the respective gene from `dat`. For the corresponding values of each row (gene), see the description of the function `wasserstein.sc`, where the argument `inclZero=TRUE` in `.testWass` has to be identified with the argument `method="OS"`, and the argument `inclZero=FALSE` with the argument `method="TS"`.

References

Schefzik, R., Flesch, J., and Goncalves, A. (2021). Fast identification of differential distributions in single-cell RNA-sequencing data with waddR.

`.wassersteinTestAsy` *Asymptotic theory-based test using the 2-Wasserstein distance to check for differential distributions*

Description

Two-sample test to check for differences between two distributions using the 2-Wasserstein distance: Implementation using a test based on asymptotic theory

Usage

```
.wassersteinTestAsy(x, y)
```

Arguments

x	sample (vector) representing the distribution of condition A
y	sample (vector) representing the distribution of condition B

Details

This is the asymptotic version of `wasserstein.test`, for the semi-parametric procedure see `.wassersteinTestSp`.

Details concerning the testing procedure based on asymptotic theory can be found in Schefzik et al (2020).

Note that the asymptotic theory-based test should only be employed when the two samples x and y can be assumed to come from continuous distributions.

Value

A vector of 13, see Schefzik et al. (2020) for details:

- d.wass: 2-Wasserstein distance between the two samples computed by quantile approximation
- d.wass^2: squared 2-Wasserstein distance between the two samples computed by quantile approximation
- d.comp^2: squared 2-Wasserstein distance between the two samples computed by decomposition approximation
- d.comp: 2-Wasserstein distance between the two samples computed by decomposition approximation
- location: location term in the decomposition of the squared 2-Wasserstein distance between the two samples
- size: size term in the decomposition of the squared 2-Wasserstein distance between the two samples

- shape: shape term in the decomposition of the squared 2-Wasserstein distance between the two samples
- rho: correlation coefficient in the quantile-quantile plot
- pval: p-value of the 2-Wasserstein distance-based test using asymptotic theory
- perc.loc: fraction (in %) of the location part with respect to the overall squared 2-Wasserstein distance obtained by the decomposition approximation
- perc.size: fraction (in %) of the size part with respect to the overall squared 2-Wasserstein distance obtained by the decomposition approximation
- perc.shape: fraction (in %) of the shape part with respect to the overall squared 2-Wasserstein distance obtained by the decomposition approximation
- decomp.error: relative error between the squared 2-Wasserstein distance obtained by the quantile approximation and the squared 2-Wasserstein distance obtained by the decomposition approximation

References

Schefzik, R., Flesch, J., and Goncalves, A. (2020). waddR: Using the 2-Wasserstein distance to identify differences between distributions in two-sample testing, with application to single-cell RNA-sequencing data.

`.wassersteinTestSp` *Semi-parametric test using the 2-Wasserstein distance to check for differential distributions*

Description

Two-sample test to check for differences between two distributions using the 2-Wasserstein distance: Semi-parametric implementation using a permutation test with a generalized Pareto distribution (GPD) approximation to estimate small p-values accurately

Usage

```
.wassersteinTestSp(x, y, permnum = 10000)
```

Arguments

`x`	sample (vector) representing the distribution of condition A
`y`	sample (vector) representing the distribution of condition B
`permnum`	number of permutations used in the permutation testing procedure

Details

This is the semi-parametric version of `wasserstein.test`, for the asymptotic theory-based procedure see `.wassersteinTestAsy`.

Details concerning the permutation testing procedure with GPD approximation to estimate small p-values accurately can be found in Schefzik et al. (2020).

Value

A vector of 15, see Schefzik et al. (2020) for details:

- d.wass: 2-Wasserstein distance between the two samples computed by quantile approximation
- d.wass^2: squared 2-Wasserstein distance between the two samples computed by quantile approximation
- d.comp^2: squared 2-Wasserstein distance between the two samples computed by decomposition approximation
- d.comp: 2-Wasserstein distance between the two samples computed by decomposition approximation
- location: location term in the decomposition of the squared 2-Wasserstein distance between the two samples
- size: size term in the decomposition of the squared 2-Wasserstein distance between the two samples
- shape: shape term in the decomposition of the squared 2-Wasserstein distance between the two samples
- rho: correlation coefficient in the quantile-quantile plot
- pval: p-value of the semi-parametric 2-Wasserstein distance-based test
- p.ad.gpd: in case the GPD fitting is performed: p-value of the Anderson-Darling test to check whether the GPD actually fits the data well (otherwise NA).
- N.exc: in case the GPD fitting is performed: number of exceedances (starting with 250 and iteratively decreased by 10 if necessary) that are required to obtain a good GPD fit, i.e. p-value of Anderson-Darling test ≥ 0.05 (otherwise NA).
- perc.loc: fraction (in %) of the location part with respect to the overall squared 2-Wasserstein distance obtained by the decomposition approximation
- perc.size: fraction (in %) of the size part with respect to the overall squared 2-Wasserstein distance obtained by the decomposition approximation
- perc.shape: fraction (in %) of the shape part with respect to the overall squared 2-Wasserstein distance obtained by the decomposition approximation
- decomp.error: relative error between the squared 2-Wasserstein distance obtained by the quantile approximation and the squared 2-Wasserstein distance obtained by the decomposition approximation

References

Schefzik, R., Flesch, J., and Goncalves, A. (2020). waddR: Using the 2-Wasserstein distance to identify differences between distributions in two-sample testing, with application to single-cell RNA-sequencing data.

.wassPermProcedure *Permutation procedure that calculates the squared 2-Wasserstein distances for random shuffles of two input samples representing two conditions*

Description

Permutation procedure that calculates the squared 2-Wasserstein distances for random shuffles of two input samples representing two conditions A and B, respectively (i.e. the elements of the pooled sample vector are randomly assigned to either condition A or condition B, where the sample sizes are as in the original samples).

Usage

```
.wassPermProcedure(x, y, permnum)
```

Arguments

x	sample (vector) representing the distribution of condition A
y	sample (vector) representing the distribution of condition B
permnum	number of permutations to be performed

Value

Vector with squared 2-Wasserstein distances computed for random shuffles of the two input samples

brownianBridgeEmpcdf.url *URL for downloading the asymptotic CDF occuring in the asymptotic theory-based test*

Description

URL for downloading the cumulative distribution function (CDF) of the integral over the squared standard Brownian bridge in the unit interval, where the computation is based on Monte Carlo simulations. This CDF occurs as an asymptotic distribution in the asymptotic theory-based test using the 2-Wasserstein distance, see Schefzik et al. (2020) for details. It is used to determine the corresponding p-values in the function `.wassersteinTestAsy`.

Usage

```
brownianBridgeEmpcdf.url
```

Format

An object of class `character` of length 1.

References

Schefzik, R., Flesch, J., and Goncalves, A. (2020). waddR: Using the 2-Wasserstein distance to identify differences between distributions in two-sample testing, with application to single-cell RNA-sequencing data.

`permutations` *Return permutations of a given vector as columns in a matrix*

Description

Returns permutations of a given vector as columns in a matrix

Usage

```
permutations(x, num_permutations)
```

Arguments

`x`	vector that is to be permutated
`num_permutations`	number of permutations to be performed

Value

a matrix, where each of the `num_permutations` columns represents one permutation of the input vector

Examples

```
x <- 1:10
set.seed(24)
permutations(x, 5)
```

`squared_wass_approx` *Compute approximated squared 2-Wasserstein distance*

Description

Calculates an approximated squared 2-Wasserstein distance based on the mean squared difference between 1000 equidistant quantiles corresponding to the empirical distributions of two input vectors x and y

Usage

```
squared_wass_approx(x, y)
```

Arguments

x	sample (vector) representing the distribution of condition A
y	sample (vector) representing the distribution of condition B

Value

The approximated squared 2-Wasserstein distance between x and y

References

Schefzik, R., Flesch, J., and Goncalves, A. (2020). waddR: Using the 2-Wasserstein distance to identify differences between distributions in two-sample testing, with application to single-cell RNA-sequencing data.

See Also

See the functions `wasserstein_metric` and `squared_wass_decomp` for alternative implementations of the 2-Wasserstein distance

Examples

```
set.seed(24)
x<-rnorm(100)
y1<-rnorm(150)
y2<-rexp(150,3)
y3<-rpois(150,2)

squared_wass_approx(x,y1)
squared_wass_approx(x,y2)
squared_wass_approx(x,y3)
```

`squared_wass_decomp` *Compute the squared 2-Wasserstein distance based on a decomposition*

Description

Computes the squared 2-Wasserstein distance between two vectors based on a decomposition into location, size and shape terms. For a detailed description of the (empirical) calculation of the invoved quantities, see Schefzik et al. (2020).

Usage

```
squared_wass_decomp(x, y)
```

Arguments

x	sample (vector) representing the distribution of condition A
y	sample (vector) representing the distribution of condition B

Value

A list of 4:

- distance: the sum location+size+shape
- location: location part in the decoposition of the 2-Wasserstein distance
- size: size part in the decoposition of the 2-Wasserstein distance
- shape: shape part in the decoposition of the 2-Wasserstein distance

References

Schefzik, R., Flesch, J., and Goncalves, A. (2020). waddR: Using the 2-Wasserstein distance to identify differences between distributions in two-sample testing, with application to single-cell RNA-sequencing data.

See Also

See the functions `wasserstein_metric` and `squared_wass_approx` for alternative implementations of the 2-Wasserstein distance

Examples

```
set.seed(24)
x<-rnorm(100)
y1<-rnorm(150)
y2<-rexp(150,3)
y3<-rpois(150,2)

squared_wass_decomp(x,y1)
squared_wass_decomp(x,y2)
squared_wass_decomp(x,y3)
```

---

testZeroes *Test for differential proportions of zero gene expression*

Description

Test for differential proportions of zero expression between two conditions for a specified set of genes; adapted from the R/Bioconductor package `scDD` by Korthauer et al. (2016)

Usage

```
testZeroes(x, y, these = seq_len(nrow(x)))

## S4 method for signature 'matrix,vector,ANY'
testZeroes(x, y, these = seq_len(nrow(x)))

## S4 method for signature 'SingleCellExperiment,SingleCellExperiment,vector'
testZeroes(x, y, these = seq_len(nrow(x)))
```

Arguments

x	matrix of single-cell RNA-sequencing expression data with genes in rows and cells (samples) in columns [alternatively, a SingleCellExperiment object for condition A, where the matrix of the single-cell RNA sequencing expression data has to be supplied via the counts argument in SingleCellExperiment]
y	vector of condition labels [alternatively, a SingleCellExperiment object for condition B, where the matrix of the single-cell RNA sequencing expression data has to be supplied via the counts argument in SingleCellExperiment]
these	vector of row numbers (i.e. gene numbers) employed to test for differential proportions of zero expression; default is seq_len(nrow(dat))

Details

Test for differential proportions of zero gene expression between two conditions using a logistic regression model accounting for the cellular detection rate. Adapted from the R/Bioconductor package scDD by Korthauer et al. (2016).

In the test, the null hypothesis that there are no differential proportions of zero gene expression (DPZ) is tested against the alternative that there are DPZ.

Value

A vector of (unadjusted) p-values

References

Korthauer, K. D., Chu, L.-F., Newton, M. A., Li, Y., Thomson, J., Stewart, R., and Kendziorski, C. (2016). A statistical approach for identifying differential distributions in single-cell RNA-seq experiments. Genome Biology, 17:222.

Examples

```
#simulate scRNA-seq data
set.seed(24)
nb.sim1<-rnbinom(n=(750*250),1,0.7)
dat1<-matrix(data=nb.sim1,nrow=750,ncol=250)
nb.sim2a<-rnbinom(n=(250*100),1,0.7)
dat2a<-matrix(data=nb.sim2a,nrow=250,ncol=100)
nb.sim2b<-rnbinom(n=(250*150),5,0.2)
dat2b<-matrix(data=nb.sim2b,nrow=250,ncol=150)
```

```
dat2<-cbind(dat2a,dat2b)
dat<-rbind(dat1,dat2)*0.25
#randomly shuffle the rows of the matrix to create the input matrix
set.seed(32)
dat<-dat[sample(nrow(dat)),]
condition<-c(rep("A",100),rep("B",150))

#call testZeroes with a matrix and a vector including conditions
#test for differential proportions of zero expression over all rows (genes)
testZeroes(dat, condition)
#test for differential proportions of zero expression only for the second row (gene)
testZeroes(dat, condition, these=2)

#alternatively, call testZeroes with two SingleCellExperiment objects
#note that the possibly pre-processed and normalized expression matrices need to be
#included using the "counts" argument
sce.A <- SingleCellExperiment::SingleCellExperiment(
  assays=list(counts=dat[,1:100]))
sce.B <- SingleCellExperiment::SingleCellExperiment(
  assays=list(counts=dat[,101:250]))
#test for differential proportions of zero expression over all rows (genes)
testZeroes(sce.A,sce.B,these=seq_len(nrow(sce.A)))
#test for differential proportions of zero expression only for the second row (gene)
testZeroes(sce.A,sce.B,these=2)
```

`wasserstein.sc` *Two-sample semi-parametric test for single-cell RNA-sequencing data to check for differences between two distributions using the 2-Wasserstein distance*

Description

Two-sample test for single-cell RNA-sequencing data to check for differences between two distributions using the 2-Wasserstein distance: Semi-parametric implementation using a permutation test with a generalized Pareto distribution (GPD) approximation to estimate small p-values accurately

Usage

```
wasserstein.sc(x, y, method = c("TS", "OS"), permnum = 10000, seed = NULL)

## S4 method for signature 'matrix,vector'
wasserstein.sc(x, y, method = c("TS", "OS"), permnum = 10000, seed = NULL)

## S4 method for signature 'SingleCellExperiment,SingleCellExperiment'
wasserstein.sc(x, y, method = c("TS", "OS"), permnum = 10000, seed = NULL)
```

Arguments

x	matrix of single-cell RNA-sequencing expression data with genes in rows and cells (samples) in columns [alternatively, a SingleCellExperiment object for condition A, where the matrix of the single-cell RNA sequencing expression data has to be supplied via the counts argument in SingleCellExperiment]
y	vector of condition labels [alternatively, a SingleCellExperiment object for condition B, where the matrix of the single-cell RNA sequencing expression data has to be supplied via the counts argument in SingleCellExperiment]
method	method employed in the testing procedure: if "OS", a one-stage test is performed, i.e. the semi-parametric test is applied to all (zero and non-zero) expression values; if "TS", a two-stage test is performed, i.e. the semi-parametric test is applied to non-zero expression values only and combined with a separate test for differential proportions of zero expression using logistic regression
permnum	number of permutations used in the permutation testing procedure
seed	number to be used to generate a L'Ecuyer-CMRG seed, which itself seeds the generation of an nextRNGStream() for each gene to achieve reproducibility; default is NULL, and no seed is set

Details

Details concerning the testing procedure for single-cell RNA-sequencing data can be found in Schefzik et al. (2020). Corresponds to the function .testWass when identifying the argument inclZero=TRUE in .testWass with the argument method="OS" and the argument inclZero=FALSE with the argument method="TS".

The input data matrix x [alternatively, the input data matrices to form the SingleCellExperiment objects x and y, respectively] as the starting point of the test is supposed to contain the single-cell RNA-sequencing expression data after several pre-processing steps. In particular, note that as input for scRNA-seq analysis, waddR expects a table of pre-filtered and normalised count data. As filtering and normalisation are important steps that can have a profound impact in a scRNA-seq workflow (Cole et al., 2019), these should be tailored to the specific question of interest before applying waddR. waddR is applicable to data from any scRNA-seq platform (demonstrated in our paper for 10x Genomics and Fluidigm C1 Smart-Seq2) normalised using most common methods, such as those implemented in the Seurat (Butler et al., 2018) or scran (Lun et al., 2016) packages. Normalisation approaches that change the shape of the gene distributions (such as quantile normalisation) and gene-wise scaling or standardizing should be avoided when using waddR.

For the two-stage approach (method="TS") according to Schefzik et al. (2021), two separate tests for differential proportions of zero expression (DPZ) and non-zero differential distributions (non-zero DD), respectively, are performed. In the DPZ test using logistic regression, the null hypothesis that there are no DPZ is tested against the alternative that there are DPZ. In the non-zero DD test using the semi-parametric 2-Wasserstein distance-based procedure, the null hypothesis that there is no difference in the non-zero expression distributions is tested against the alternative that the two non-zero expression distributions are differential.

The current implementation of the test assumes that the expression data matrix is based on one replicate per condition only. For approaches on how to address settings comprising multiple replicates per condition, see Schefzik et al. (2021).

Value

Matrix, where each row contains the testing results of the respective gene from `dat`. The corresponding values of each row (gene) are as follows, see Schefzik et al. (2021) for details. In case of `inclZero=TRUE`:

- d.wass: 2-Wasserstein distance between the two samples computed by quantile approximation
- d.wass^2: squared 2-Wasserstein distance between the two samples computed by quantile approximation
- d.comp^2: squared 2-Wasserstein distance between the two samples computed by decomposition approximation
- d.comp: 2-Wasserstein distance between the two samples computed by decomposition approximation
- location: location term in the decomposition of the squared 2-Wasserstein distance between the two samples
- size: size term in the decomposition of the squared 2-Wasserstein distance between the two samples
- shape: shape term in the decomposition of the squared 2-Wasserstein distance between the two samples
- rho: correlation coefficient in the quantile-quantile plot
- pval: p-value of the semi-parametric 2-Wasserstein distance-based test when applied to all (zero and non-zero) respective gene expression values
- p.ad.gpd: in case the GPD fitting is performed: p-value of the Anderson-Darling test to check whether the GPD actually fits the data well (otherwise NA)
- N.exc: in case the GPD fitting is performed: number of exceedances (starting with 250 and iteratively decreased by 10 if necessary) that are required to obtain a good GPD fit, i.e. p-value of Anderson-Darling test ≥ 0.05 (otherwise NA)
- perc.loc: fraction (in %) of the location part with respect to the overall squared 2-Wasserstein distance obtained by the decomposition approximation
- perc.size: fraction (in %) of the size part with respect to the overall squared 2-Wasserstein distance obtained by the decomposition approximation
- perc.shape: fraction (in %) of the shape part with respect to the overall squared 2-Wasserstein distance obtained by the decomposition approximation
- decomp.error: relative error between the squared 2-Wasserstein distance computed by the quantile approximation and the squared 2-Wasserstein distance computed by the decomposition approximation
- pval.adj: adjusted p-value of the semi-parametric 2-Wasserstein distance-based test according to the method of Benjamini-Hochberg (i.e. adjusted p-value corresponding to pval)

In case of `inclZero=FALSE`:

- d.wass: 2-Wasserstein distance between the two samples computed by quantile approximation (based on non-zero expression only)
- d.wass^2: squared 2-Wasserstein distance between the two samples computed by quantile approximation (based on non-zero expression only)

- d.comp^2: squared 2-Wasserstein distance between the two samples computed by decomposition approximation (based on non-zero expression only)
- d.comp: 2-Wasserstein distance between the two samples computed by decomposition approximation (based on non-zero expression only)
- location: location term in the decomposition of the squared 2-Wasserstein distance between the two samples (based on non-zero expression only)
- size: size term in the decomposition of the squared 2-Wasserstein distance between the two samples (based on non-zero expression only)
- shape: shape term in the decomposition of the squared 2-Wasserstein distance between the two samples (based on non-zero expression only)
- rho: correlation coefficient in the quantile-quantile plot (based on non-zero expression only)
- p.nonzero: p-value of the semi-parametric 2-Wasserstein distance-based test when applied to non-zero respective gene expression values
- p.ad.gpd: in case the GPD fitting is performed: p-value of the Anderson-Darling test to check whether the GPD actually fits the data well (otherwise NA)
- N.exc: in case the GPD fitting is performed: number of exceedances (starting with 250 and iteratively decreased by 10 if necessary) that are required to obtain a good GPD fit, i.e. p-value of Anderson-Darling test ≥ 0.05 (otherwise NA)
- perc.loc: fraction (in %) of the location part with respect to the overall squared 2-Wasserstein distance obtained by the decomposition approximation (based on non-zero expression only)
- perc.size: fraction (in %) of the size part with respect to the overall squared 2-Wasserstein distance obtained by the decomposition approximation (based on non-zero expression only)
- perc.shape: fraction (in %) of the shape part with respect to the overall squared 2-Wasserstein distance obtained by the decomposition approximation (based on non-zero expression only)
- decomp.error: relative error between the squared 2-Wasserstein distance computed by the quantile approximation and the squared 2-Wasserstein distance computed by the decomposition approximation (based on non-zero expression only)
- p.zero: p-value of the test for differential proportions of zero expression (logistic regression model)
- p.combined: combined p-value of p.nonzero and p.zero obtained by Fisher's method
- p.adj.nonzero: adjusted p-value of the semi-parametric 2-Wasserstein distance-based test based on non-zero expression only according to the method of Benjamini-Hochberg (i.e. adjusted p-value corresponding to p.nonzero)
- p.adj.zero: adjusted p-value of the test for differential proportions of zero expression (logistic regression model) according to the method of Benjamini-Hochberg (i.e. adjusted p-value corresponding to p.zero)
- p.adj.combined: adjusted combined p-value of p.nonzero and p.zero obtained by Fisher's method according to the method of Benjamini-Hochberg (i.e. adjusted p-value corresponding to p.combined)

References

Butler, A., Hoffman, P., Smibert, P., Papalexi, E., and Satija, R. (2018). Integrating single-cell transcriptomic data across different conditions, technologies, and species. Nature Biotechnology, 36, 411-420.

Cole, M. B., Risso, D., Wagner, A., De Tomaso, D., Ngai, J., Purdom, E., Dudoit, S., and Yosef, N. (2019). Performance assessment and selection of normalization procedures for single-cell RNA-seq. Cell Systems, 8, 315-328.

Lun, A. T. L., Bach, K., and Marioni, J. C. (2016). Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. Genome Biology, 17, 75.

Schefzik, R., Flesch, J., and Goncalves, A. (2021). Fast identification of differential distributions in single-cell RNA-sequencing data with waddR.

Examples

```
#simulate scRNA-seq data
set.seed(24)
nb.sim1<-rnbinom(n=(750*250),1,0.7)
dat1<-matrix(data=nb.sim1,nrow=750,ncol=250)
nb.sim2a<-rnbinom(n=(250*100),1,0.7)
dat2a<-matrix(data=nb.sim2a,nrow=250,ncol=100)
nb.sim2b<-rnbinom(n=(250*150),5,0.2)
dat2b<-matrix(data=nb.sim2b,nrow=250,ncol=150)
dat2<-cbind(dat2a,dat2b)
dat<-rbind(dat1,dat2)*0.25
#randomly shuffle the rows of the matrix to create the input matrix
set.seed(32)
dat<-dat[sample(nrow(dat)),]
condition<-c(rep("A",100),rep("B",150))

#call wasserstein.sc with a matrix and a vector including conditions
#set seed for reproducibility
#two-stage method
wasserstein.sc(dat,condition,method="TS",permnum=10000,seed=24)
#one-stage method
wasserstein.sc(dat,condition,method="OS",permnum=10000,seed=24)

#alternatively, call wasserstein.sc with two SingleCellExperiment objects
#note that the possibly pre-processed and normalized expression matrices need to be
#included using the "counts" argument
sce.A <- SingleCellExperiment::SingleCellExperiment(
  assays=list(counts=dat[,1:100]))
sce.B <- SingleCellExperiment::SingleCellExperiment(
  assays=list(counts=dat[,101:250]))
#set seed for reproducibility
#two-stage method
wasserstein.sc(sce.A,sce.B,method="TS",permnum=10000,seed=24)
#one-stage method
wasserstein.sc(sce.A,sce.B,method="OS",permnum=10000,seed=24)
```

`wasserstein.test` *Two-sample test to check for differences between two distributions using the 2-Wasserstein distance*

Description

Two-sample test to check for differences between two distributions using the 2-Wasserstein distance, either using the semi-parametric permutation testing procedure with a generalized Pareto distribution (GPD) approximation to estimate small p-values accurately or the test based on asymptotic theory

Usage

```
wasserstein.test(x, y, method = c("SP", "ASY"), permnum = 10000)
```

Arguments

x	sample (vector) representing the distribution of condition A
y	sample (vector) representing the distribution of condition B
method	testing procedure to be employed: "SP" for the semi-parametric permutation testing procedure with GPD approximation, "ASY" for the test based on asymptotic theory; if no method is specified, "SP" will be used by default.
permnum	number of permutations used in the permutation testing procedure (if `method="SP"` is performed); default is 10000

Details

Details concerning the two testing procedures (i.e. the semi-parametric permutation testing procedure with GPD approximation and the test based on asymptotic theory) can be found in Schefzik et al. (2020).

Note that the asymptotic theory-based test (`method="ASY"`) should only be employed when the samples x and y can be assumed to come from continuous distributions. In contrast, the semi-parametric test (`method="SP"`) can be used for samples coming from continuous or discrete distributions.

Value

A vector, see Schefzik et al. (2020) for details:

- d.wass: 2-Wasserstein distance between the two samples computed by quantile approximation
- d.wass^2: squared 2-Wasserstein distance between the two samples computed by quantile approximation
- d.comp^2: squared 2-Wasserstein distance between the two samples computed by decomposition approximation
- d.comp: 2-Wasserstein distance between the two samples computed by decomposition approximation
- location: location term in the decomposition of the squared 2-Wasserstein distance between the two samples
- size: size term in the decomposition of the squared 2-Wasserstein distance between the two samples
- shape: shape term in the decomposition of the squared 2-Wasserstein distance between the two samples

- rho: correlation coefficient in the quantile-quantile plot
- pval: The p-value of the semi-parametric or the asymptotic theory-based test, depending on the specified method
- p.ad.gpd: in case the GPD fitting is performed: p-value of the Anderson-Darling test to check whether the GPD actually fits the data well (otherwise NA). This output is only returned when performing the semi-parametric test (method="SP")!
- N.exc: in case the GPD fitting is performed: number of exceedances (starting with 250 and iteratively decreased by 10 if necessary) that are required to obtain a good GPD fit, i.e. p-value of Anderson-Darling test ≥ 0.05 (otherwise NA). This output is only returned when performing the semi-parametric test (method="SP")!
- perc.loc: fraction (in %) of the location part with respect to the overall squared 2-Wasserstein distance obtained by the decomposition approximation
- perc.size: fraction (in %) of the size part with respect to the overall squared 2-Wasserstein distance obtained by the decomposition approximation
- perc.shape: fraction (in %) of the shape part with respect to the overall squared 2-Wasserstein distance obtained by the decomposition approximation
- decomp.error: relative error between the squared 2-Wasserstein distance computed by the quantile approximation and the squared 2-Wasserstein distance computed by the decomposition approximation

References

Schefzik, R., Flesch, J., and Goncalves, A. (2020). waddR: Using the 2-Wasserstein distance to identify differences between distributions in two-sample testing, with application to single-cell RNA-sequencing data.

Examples

```
set.seed(24)
x<-rnorm(100)
y1<-rnorm(150)
y2<-rexp(150,3)
y3<-rpois(150,2)

#for reproducibility, set a seed for the semi-parametric, permutation-based test
set.seed(32)
wasserstein.test(x,y1,method="SP",permnum=10000)
wasserstein.test(x,y1,method="ASY")

set.seed(33)
wasserstein.test(x,y2,method="SP",permnum=10000)
wasserstein.test(x,y2,method="ASY")

set.seed(34)
#only consider SP method, as Poisson distribution is discrete
wasserstein.test(x,y3,method="SP",permnum=10000)
```

`wasserstein_metric` *Calculate the p-Wasserstein distance*

Description

Calculates the p-Wasserstein distance (metric) between two vectors x and y

Usage

```
wasserstein_metric(x, y, p = 1, wa_ = NULL, wb_ = NULL)
```

Arguments

`x`	sample (vector) representing the distribution of condition A
`y`	sample (vector) representing the distribution of condition B
`p`	order of the Wasserstein distance
`wa_`	optional vector of weights for `x`
`wb_`	optional vector of weights for `y`

Details

This implementation of the p-Wasserstein distance is a Rcpp reimplementation of the `wasserstein1d` function from the R package `transport` by Schuhmacher et al.

Value

The p-Wasserstein distance between x and y

References

Schefzik, R., Flesch, J., and Goncalves, A. (2020). waddR: Using the 2-Wasserstein distance to identify differences between distributions in two-sample testing, with application to single-cell RNA-sequencing data.

See Also

See the functions `squared_wass_approx` and `squared_wass_decomp` for alternative implementations of the 2-Wasserstein distance.

Examples

```
set.seed(24)
x<-rnorm(100)
y1<-rnorm(150)
y2<-rexp(150,3)
y3<-rpois(150,2)
```

```
#calculate 2-Wasserstein distance between x and y1
wasserstein_metric(x,y1,p=2)
#calculate squared 2-Wasserstein distance between x and y1
wasserstein_metric(x,y1,p=2)^2

#calculate 2-Wasserstein distance between x and y2
wasserstein_metric(x,y2,p=2)
#calculate squared 2-Wasserstein distance between x and y2
wasserstein_metric(x,y2,p=2)^2

#calculate 2-Wasserstein distance between x and y3
wasserstein_metric(x,y3,p=2)
#calculate squared 2-Wasserstein distance between x and y3
wasserstein_metric(x,y3,p=2)^2
```

Index

∗ **datasets**
    brownianBridgeEmpcdf.url, 12
.brownianBridgeEmpcdf, 4
.combinePVal, 5
.fishersCombinedPval, 5
.gpdFittedPValue, 6
.quantileCorrelation, 7
.relativeError, 7
.testWass, 8
.wassPermProcedure, 12
.wassersteinTestAsy, 9
.wassersteinTestSp, 10

brownianBridgeEmpcdf.url, 12

permutations, 13

squared_wass_approx, 13
squared_wass_decomp, 14

testZeroes, 15
testZeroes,matrix,vector,ANY-method
        *(*testZeroes*)*, 15
testZeroes,SingleCellExperiment,SingleCellExperiment,vector-method
        *(*testZeroes*)*, 15

waddR *(*waddR-package*)*, 2
waddR-package, 2
wasserstein.sc, 17
wasserstein.sc,matrix,vector-method
        *(*wasserstein.sc*)*, 17
wasserstein.sc,SingleCellExperiment,SingleCellExperiment,ANY,ANY,ANY-method
        *(*wasserstein.sc*)*, 17
wasserstein.sc,SingleCellExperiment,SingleCellExperiment-method
        *(*wasserstein.sc*)*, 17
wasserstein.sc-method,matrix,vector,ANY,ANY,ANY-method
        *(*wasserstein.sc*)*, 17
wasserstein.test, 21
wasserstein_metric, 24