Package 'waddR'

March 30, 2021

Type Package

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

Version 1.4.0

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 welll as 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), BiocFileCache, BiocParallel, SingleCellExperiment, parallel, methods, stats

Depends R (>= 3.6.0)

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

LinkingTo Rcpp, RcppArmadillo,

VignetteBuilder knitr

RoxygenNote 7.0.2

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

git_branch RELEASE_3_12

git_last_commit 8e4b97d

git_last_commit_date 2020-10-27

Date/Publication 2021-03-29

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

Maintainer Julian Flesch <julianflesch@gmail.com>

R topics documented:

`waddR-package` *waddR: Statistical Test for Detecting Differential Distributions Based on the Wasserstein Distance*

Description

Wasserstein distance based statistical test for detecting and describing differential distributions in one-dimensional data. Functions for wasserstein distance calculation, differential distribution testing, and a specialized test for differential expression in scRNA data are provided.

Details

The Wasserstein package offers utilities for three distinct use cases:

- Computation of the 2-Wasserstein distance
- Two-sample test to check for differences between two distributions
- Detect differential gene expression distributions in scRNAseq data

Wasserstein Distance functions

The 2-Wasserstein distance is a metric to describe the distance between two distributions, representing two diferent conditions A and B. This package specifically considers the squared 2-Wasserstein distance d := W^2 which offers a decomposition into location, size, and shape terms. It offers three functions to calculate the 2-Wasserstein distance, all of which are implemented in Cpp and exported to R with Rcpp for better performance. `wasserstein_metric` is a Cpp reimplementation of the wasserstein1d method from the package `transport` and offers the most exact results. The functions `squared_wass_approx` and `squared_wass_decomp` compute approximations of the squared 2-Wasserstein distance with `suared_wass_decomp` also returning the decomosition terms for location, size, and shape. See `?wasserstein_metric`, `?squared_wass_aprox`, and `?squared_wass_decomp` as well as the accompanying paper Schefzik and Goncalves 2019.

Two-Sample Testing

This 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 ba specifically testing the null hypothesis H0: F_A = F_B against the alternative hypothesis H1: F_A != F_B.

The first, semi-parametric (SP), procedure uses a test based on permutations combined with a generalized pareto distribution approximation to estimate small pvalues accurately.

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

See the documentation of the function `?wasserstein.test` for more details.

Single Cell Test

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 single-cell RNA-seqencing (scRNA-seq) data. In particular, a two-stage (TS) approach has been 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 the documentation of the Single Cell testing function `?wasserstein.sc` and the test for zero expression levels `?testZeroes` for more details.

Author(s)

Maintainer: Julian Flesch <julianflesch@gmail.com>

Authors:

- Roman Schefzik <r.schefzik@dkfz-heidelberg.de>

See Also

Useful links:

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

---

`.brownianBridgeEmpcdf` *.brownianBridgeEmpcdf*

Description

An empirical cumulative distribution function of a simulated Brownian bridge distribution. It is used an empirical quantile function to determine p-values in the asymptotic wasserstein test function .wassersteinTestAsy

Usage

```
.brownianBridgeEmpcdf(v)
```

Arguments

v	distribution funtion input value

Value

Value at x of the Brownian distribution

.combinePVal *.combinePVal*

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

Value

A vector of length N of the combined p-values

.fishersCombinedPval *.fishersCombinedPval*

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

.quantileCorrelation *.quantileCorrelation*

Description

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

Usage

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

Arguments

x	numeric vector representing distribution x
y	numeric vector representing distribution y
pr	probabilities for computing the quantiles of x and y. By default is set to 1000 equidistant quantiles starting at q_1 = 0.5/1000.

Value

quantile-quantile correlation of x and y

.relativeError *.relativeError*

Description

Computes the relative error between two numericals (in the sense of is.numeric) x and y. If x and y are vectors, it is assumed that length(x) == length(y).

Usage

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

Arguments

x	numerical (in the sense of is.numeric)
y	numerical (in the sense of is.numeric)

Details

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

Value

The relative error between x and y

`.testWass` *.testWass*

Description

Two-sample test for single-cell RNA-sequencing data to check for differences between two distributions (conditions) 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 genes in rows and samples (cells) in columns
`condition`	vector of condition labels
`permnum`	number of permutations used in the permutation testing procedure
`inclZero`	logical; if TRUE, the one-stage method (i.e. semi-parametric testing applied to all (zero and non-zero) expression values) is performed; if FALSE, the two-stage method (i.e. semi-parametric testing applied to non-zero expression values only, combined with a separate testing for differential proportions of zero expression using logistic regression) is performed. 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. By default, NULL is given and no seed is set.

Details

Details concerning the permutation testing procedures for single-cell RNA-sequencing data can be found in Schefzik and Goncalves (2019).

Value

matrix with every row being the wasserstein test of one gene between the two conditions. See the corresponding values in 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 in .testWass with the argument method="TS".

References

Schefzik and Goncalves 2019

`.wassersteinTestAsy` *.wassersteinTestAsy*

Description

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

Usage

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

Arguments

x	univariate sample (vector) representing the distribution of condition A
y	univariate 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 and Goncalves (2019).

Value

A vector concerning the testing results, precisely (see Schefzik and Goncalves (2019) 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 overall squared 2-Wasserstein distance obtained by the decomposition approximation
- perc.size: fraction (in overall squared 2-Wasserstein distance obtained by the decomposition approximation

- perc.shape: fraction (in 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. and Goncalves, A. (2019).

`.wassersteinTestSp` *.wassersteinTestSp*

Description

Two-sample test to check for differences between two distributions (conditions) 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`	univariate sample (vector) representing the distribution of condition A
`y`	univariate 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 procedure see .wassersteinTestAsy

Details concerning the permutation testing procedure with GPD approximation to estimate small p-values accurately can be found in Schefzik and Goncalves (2019).

Value

A vector concerning the testing results, precisely (see Schefzik and Goncalves (2019) 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). NOTE: GPD fitting is currently not supported!
- 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 greater or eqaul to 0.05) (otherwise NA) NOTE: GPD fitting is currently not supported!
- perc.loc: fraction (in overall squared 2-Wasserstein distance obtained by the decomposition approximation
- perc.size: fraction (in overall squared 2-Wasserstein distance obtained by the decomposition approximation
- perc.shape: fraction (in 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. and Goncalves, A. (2019).

`.wassPermProcedure` *.wassPermProcedure*

Description

Permutation Procedure that calculates the squared 2-Wasserstein distance for random shuffles of two input distributions and returns them as a vector.

Usage

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

Arguments

`x`	numeric vector representing distribution A
`y`	numeric vector representing distribution B
`permnum`	integer interpreted as the number of permutations to be performed

Value

vector with squared 2-Wasserstein distances computed on random shuffles of the two input vectors

brownianBridgeEmpcdf.url *brownianBridgeEmpcdf.url*

Description

Url for downloading the simulated Brownian bridge distribution. It is used an empirical quantile function to determine p-values in the asymptotic wasserstein test function .wassersteinTestAsy

Usage

```
brownianBridgeEmpcdf.url
```

Format

An object of class `character` of length 1.

permutations *permutations*

Description

Returns permutations of a given NumericVector as columns in a NumericMatrix object.

Usage

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

Arguments

`x`	NumericVector representing a vector that is to be permutated
`num_permutations`	Integer representing the number of permutations that are to be performed.

Value

a matrix containing in every column one permutations of the input vector

Examples

```
x <- seq(1:10)
m <- permutations(x, 5)
dim(m)
#[1] 10  5
```

squared_wass_approx *squared_wass_approx*

Description

Approximation of the squared wasserstein distance. Calculation based on the mean squared difference between the equidistant empirical quantiles of the two input vectors a and b. As an approximation of the quantile function, 1000 quantiles are computed for each vector.

Usage

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

Arguments

x	Vector representing an empirical distribution under condition A
y	Vector representing an empirical distribution under condition B

Value

The approximated squared wasserstein distance between x and y

References

Schefzik and Goncalves 2019

See Also

[wasserstein_metric()], [squared_wass_decomp()] for different implementations of the wasserstein distance

Examples

```
# input: one dimensional data in two conditions
x <- rnorm(100, 42, 2)
y <- c(rnorm(61, 20, 1), rnorm(41, 40,2))
# output: The squared Wasserstein distance approximated as described in
# Schefzik and Goncalves 2019
d.wass.approx <- squared_wass_approx(x,y)
```

squared_wass_decomp *squared_wass_decomp*

Description

Approximation of the squared Wasserstein distance d_g between two vectors decomposed into size, location and shape. Calculation based on the mean squared difference between the equidistant quantiles of the two input vectors a and b. As an approximation of the distribution, 1000 quantiles are computed for each vector.

Usage

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

Arguments

x	Vector representing an empirical distribution under condition A
y	Vector representing an empirical distribution under condition B

Value

An named Rcpp::List with the wasserstein distance between x and y, decomposed into terms for size, location, and shape

References

Schefzik and Goncalves 2019 Irpino and Verde (2015)

See Also

[wasserstein_metric()], [squared_wass_approx()] for different implementations of the wasserstein distance

Examples

```
# input: one dimensional data in two conditions
x <- rnorm(100, 42, 2)
y <- c(rnorm(61, 20, 1), rnorm(41, 40,2))
# output: squared Wasserstein distance decomposed into terms for location,
# shape, size
d.wass.decomp <- squared_wass_decomp(x,y)
d.wass.decomp$location
d.wass.decomp$size
d.wass.decomp$shape
```

testZeroes *testZeroes: Test for differential proportions of zero values*

Description

Test for differential proportions of zero expression between two conditions for a specified set of genes

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 samples (cells) in columns
y	vector of condition labels
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 expression between two conditions that is not explained by the detection rate using a (Bayesian) logistic regression model. Adapted from the scDD package (Korthauer et al. 2016).

Value

A vector of (unadjusted) p-values

References

Korthauer et al. (2016).

Examples

```
x1 <- c(rnorm(100,42,1), rnorm(102,45,3))
x2 <- c(rnorm(100,0,1), rnorm(102,0,2))
dat <- matrix(c(x1,x2), nrow=2, byrow=TRUE)
condition <- c(rep(1,100), rep(2,102))
# test over all rows
testZeroes(dat, condition)
# only consider the second row
testZeroes(dat, condition, these=c(2))
```

wasserstein.sc *wasserstein.sc*

Description

Two-sample test for single-cell RNA-sequencing data to check for differences between two distributions (conditions) 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 samples (cells) in columns
y	vector of condition labels
method	method employed in the testing procedure: "OS" for the one-stage method (i.e. semi-parametric testing applied to all (zero and non-zero) expression values); "TS" for the two-stage method (i.e. semi-parametric testing applied to non-zero expression values only, combined with a separate testing for differential proportions of zero expression using logistic regression). If this argument is not given, a two-sided test is run by default.
permnum	number of permutations used in the permutation testing procedure. If this argument is not given, 10000 is used as default
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. By default, NULL is given and no seed is set.

Details

Details concerning the permutation testing procedures for single-cell RNA-sequencing data can be found in Schefzik and Goncalves (2019). Corresponds to the function .testWass when identifying the argument inclZero=TRUE in .testWass with the argument method="OS" and the argument inclZero=FALSE in .testWass with the argument method="TS".

Value

See the corresponding values in the description of the function .testWass, where the argument inclZero=TRUE in .testWass has to be identified with the argument method="OS", and the argument inclZero=FALSE in .testWass with the argument method="TS". A vector concerning the testing results, precisely (see Schefzik and Goncalves (2019) 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
- 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 greater or eqaul to 0.05)(otherwise NA)
- perc.loc: fraction (in overall squared 2-Wasserstein distance obtained by the decomposition approximation
- perc.size: fraction (in overall squared 2-Wasserstein distance obtained by the decomposition approximation
- perc.shape: fraction (in 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

In case of inclZero=FALSE:

- 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
- p.nonzero: p-value of the semi-parametric 2-Wasserstein distance-based test (based on non-zero expression only)
- 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 greater or eqaul to 0.05)(otherwise NA)
- perc.loc: fraction (in overall squared 2-Wasserstein distance obtained by the decomposition approximation
- perc.size: fraction (in overall squared 2-Wasserstein distance obtained by the decomposition approximation
- perc.shape: fraction (in 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
- 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
- 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
- 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

References

Schefzik and Goncalves (2019).

Examples

```
# some data in two conditions
cond1 <- matrix(rnorm(100, 42, 1), nrow=1)
cond2 <- matrix(rnorm(100, 45, 3), nrow=1)

# call wasserstein.sc with a matrix
# and a vector denoting conditions
dat <- cbind(cond1, cond2)
condition <- c(rep(1, 100), rep(2, 100))
wasserstein.sc(dat, condition, "TS", 100)

# call wasserstein.sc with two SingleCellExperiment objects
sce1 <- SingleCellExperiment::SingleCellExperiment(
          assays=list(counts=cond1, logcounts=log10(cond1)))
sce2 <- SingleCellExperiment::SingleCellExperiment(
          assays=list(counts=cond2, logcounts=log10(cond2)))
wasserstein.sc(sce1, sce2, "TS", 100)
```

```
# for reproducible p-values
wasserstein.sc(sce1, sce2, seed=123)
```

`wasserstein.test` *wasserstein.test*

Description

Two-sample test to check for differences between two distributions (conditions) using the 2-Wasserstein distance, either using the semi-parametric permutation testing procedure with 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`	univariate sample (vector) representing the distribution of condition A
`y`	univariate 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 to estimate small p-values accurately; "ASY" for the test based on asymptotic theory. If no method is given, "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 permutation testing procedure with GPD approximation to estimate small p-values accurately and the test based on asymptotic theory) can be found in Schefzik and Goncalves (2019).

Value

A vector concerning the testing results (see Schefzik and Goncalves (2019) for details).

A vector concerning the testing results, precisely (see Schefzik and Goncalves (2019) 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 2-Wasserstein distance-based test or p-value determined using asymptotic theory, depending on the 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 a 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 greater or eqaul to 0.05) (otherwise NA). This output is only returned when performing a semi-parametric test (method="SP")!
- perc.loc: fraction (in overall squared 2-Wasserstein distance obtained by the decomposition approximation
- perc.size: fraction (in overall squared 2-Wasserstein distance obtained by the decomposition approximation
- perc.shape: fraction (in 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. and Goncalves, A. (2019).

Examples

```
# generate two input distributions
x<-rnorm(500)
y<-rnorm(500,4,1.5)
wasserstein.test(x,y,method="ASY")
# Run with default options: method="SP", permnum=10000
wasserstein.test(x,y)
# Run with a seed for the semi-parametric test ("SP")
set.seed(42)
wasserstein.test(x,y, method="SP")
```

`wasserstein_metric` *wasserstein_metric*

Description

The order p Wasserstein metric (or distance) is defined as the p-th root of the total cost of turning one pile of mass x into a new pile of mass y. The cost a single transport x_i into y_i is the p-th power of the euclidean distance between x_i and y_i.

Usage

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

Arguments

x	NumericVector representing an empirical distribution under condition A
y	NumericVector representing an empirical distribution under condition B
p	order of the wasserstein distance
wa_	NumericVector representing the weights of datapoints (interpreted as clusters) in x
wb_	NumericVector representing the weights of datapoints (interpreted as clusters) in y

Details

The masses in x and y can also be represented as clusters P and Q with weights W_P and W_Q. The wasserstein distance then becomes the optimal flow F, which is the sum of all optimal flows f_{ij} from $(p_i, w_{p,i})$ to $(q_i, w_{q,i})$.

This implementation of the Wasserstein metric is a Rcpp reimplementation of the wasserstein1d function by Dominic Schuhmacher from the package transport.

Value

The wasserstein (transport) distance between x and y

References

Schefzik and Goncalves 2019

See Also

[squared_wass_approx()], [squared_wass_decomp()] for different approximations of the wasserstein distance

Examples

```
# input: one dimensional data in two conditions
x <- rnorm(100, 42, 2)
y <- c(rnorm(61, 20, 1), rnorm(41, 40, 2))
# output: The exact Wasserstein distance between the two input
# vectors. Reimplementation of the wasserstein1d function found in
# the packge transport.
d.wass <- wasserstein_metric(x,y,2)
```

Index