Package 'funStatTest'

April 20, 2023

Title Statistical Testing for Functional Data

Version 1.0.2

Description
Implementation of two sample comparison procedures based on median-based statistical tests for functional data, introduced in Smida et al (2022) <doi:10.1080/10485252.2022.2064997>. Other competitive state-of-the-art approaches proposed by Chakraborty and Chaudhuri (2015) <doi:10.1093/biomet/asu072>, Horvath et al (2013) <doi:10.1111/j.1467-9868.2012.01032.x> or Cuevas et al (2004) <doi:10.1016/j.csda.2003.10.021> are also included in the package, as well as procedures to run test result comparisons and power analysis using simulations.

License AGPL (>= 3)

URL https://plmlab.math.cnrs.fr/gdurif/funStatTest/,https://gdurif.pages.math.cnrs.fr/funStatTest/

Imports checkmate, distr, dplyr, ggplot2, magrittr, Matrix, pbapply, stats, stringr, tibble, tidyr, tidyselect, utils

Suggests knitr, rmarkdown, testthat, vdiffr

VignetteBuilder knitr

Date/Publication 2023-04-20 08:30:02 UTC

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Zaineb Smida [aut] (<https://orcid.org/0000-0002-9974-299X>), Ghislain Durif [aut, cre] (<https://orcid.org/0000-0003-2567-1401>), Lionel Cucala [aut]

Maintainer Ghislain Durif <gd.dev@libertymail.net>

Repository CRAN

R topics documented:

comp_stat 2
permut_pval 3
plot_simu 4
power_exp 5
simul_data 7
simul_traj 8
stat_cff 9
stat_hkr 10
stat_med 11
stat_mo 12
stat_wmw 13

Index **15**

`comp_stat` *Compute multiple statistics*

Description

Computation of the different statistics defined in the package. See Smida et al (2022) for more details.

Usage

```
comp_stat(MatX, MatY, stat = c("mo", "med"))
```

Arguments

`MatX`	numeric matrix of dimension `n_point x n` containing `n` trajectories (in columns) of size `n_point` (in rows).
`MatY`	numeric matrix of dimension `n_point x m` containing `m` trajectories (in columns) of size `n_point` (in rows).
`stat`	character string or vector of character string, name of the statistics for which the p-values will be computed, among `"mo"`, `"med"`, `"wmw"`, `"hkr"`, `"cff"`.

Details

For HKR statistics, only the values of the two statistics, namely `HKR1` and `HKR2` and not the eigen values (see `stat_hkr()` for more details).

Value

list of named numeric value corresponding to the statistic values listed in `stat` input.

References

Zaineb Smida, Lionel Cucala, Ali Gannoun & Ghislain Durif (2022) A median test for functional data, Journal of Nonparametric Statistics, 34:2, 520-553, doi:10.1080/10485252.2022.2064997, hal-03658578

See Also

`stat_mo()`, `stat_med()`, `stat_wmw()`, `stat_hkr()`, `stat_cff()`

Examples

```
simu_data <- simul_data(
    n_point = 100, n_obs1 = 50, n_obs2 = 75, c_val = 10,
    delta_shape = "constant", distrib = "normal"
)

MatX <- simu_data$mat_sample1
MatY <- simu_data$mat_sample2

res <- comp_stat(MatX, MatY, stat = c("mo", "med", "wmw", "hkr", "cff"))
res
```

---

permut_pval *Permutation-based computation of p-values*

Description

Computation of the p-values associated to any statistics described in the package with the permutation methods. See Smida et al (2022) for more details.

Usage

```
permut_pval(MatX, MatY, n_perm = 100, stat = c("mo", "med"), verbose = FALSE)
```

Arguments

`MatX`	numeric matrix of dimension `n_point` x `n` containing `n` trajectories (in columns) of size `n_point` (in rows).
`MatY`	numeric matrix of dimension `n_point` x `m` containing `m` trajectories (in columns) of size `n_point` (in rows).
`n_perm`	integer, number of permutation to compute the p-values.
`stat`	character string or vector of character string, name of the statistics for which the p-values will be computed, among `"mo"`, `"med"`, `"wmw"`, `"hkr"`, `"cff"`.
`verbose`	boolean, if TRUE, enable verbosity.

Value

list of named numeric value corresponding to the p-values for each statistic listed in the `stat` input.

References

Zaineb Smida, Lionel Cucala, Ali Gannoun & Ghislain Durif (2022) A median test for functional data, Journal of Nonparametric Statistics, 34:2, 520-553, doi:10.1080/10485252.2022.2064997, hal-03658578

See Also

stat_mo(), stat_med(), stat_wmw(), stat_hkr(), stat_cff(), comp_stat()

Examples

```
# simulate small data for the example
simu_data <- simul_data(
    n_point = 20, n_obs1 = 4, n_obs2 = 5, c_val = 10,
    delta_shape = "constant", distrib = "normal"
)

MatX <- simu_data$mat_sample1
MatY <- simu_data$mat_sample2
res <- permut_pval(
    MatX, MatY, n_perm = 100, stat = c("mo", "med", "wmw", "hkr", "cff"),
    verbose = TRUE)
res
```

---

plot_simu *Graphical representation of simulated data*

Description

Graphical representation of simulated data

Usage

```
plot_simu(simu)
```

Arguments

simu	list, output of simul_data()

Value

the ggplot2 graph of simulated tajectories.

See Also

simul_data()

Examples

```
# constant delta
simu_data <- simul_data(
    n_point = 100, n_obs1 = 50, n_obs2 = 75, c_val = 5,
    delta_shape = "constant", distrib = "normal"
)
plot_simu(simu_data)
# linear delta
simu_data <- simul_data(
    n_point = 100, n_obs1 = 50, n_obs2 = 75, c_val = 5,
    delta_shape = "linear", distrib = "normal"
)
plot_simu(simu_data)
# quadratic delta
simu_data <- simul_data(
    n_point = 100, n_obs1 = 50, n_obs2 = 75, c_val = 5,
    delta_shape = "quadratic", distrib = "normal"
)
plot_simu(simu_data)
```

---

power_exp *Simulation-based experiment for power analysis*

Description

Computation of the statistical power (i.e. risk to reject the null hypothesis when it is false) associated to any statistics described in the package based on simulation permutation-based p-values computations. See Smida et al (2022) for more details.

Usage

```
power_exp(
  n_simu = 100,
  alpha = 0.05,
  n_perm = 100,
  stat = c("mo", "med"),
  n_point = 100,
  n_obs1 = 50,
  n_obs2 = 50,
  c_val = 1,
  delta_shape = "constant",
  distrib = "normal",
  max_iter = 10000,
  verbose = FALSE
)
```

Arguments

n_simu	integer value, number of simulations to compute the statistical power.
alpha	numerical value, between 0 and 1, type I risk level to reject the null hypothesis in the simulation. Default value is 5%.
n_perm	integer, number of permutation to compute the p-values.
stat	character string or vector of character string, name of the statistics for which the p-values will be computed, among "mo", "med", "wmw", "hkr", "cff".
n_point	integer value, number of points in the trajectory.
n_obs1	integer value, number of trajectories in the first sample.
n_obs2	integer value, number of trajectories in the second sample.
c_val	numeric value, level of divergence between the two samples.
delta_shape	character string, shape of the divergence between the two samples, among "constant", "linear", "quadratic".
distrib	character string, type of probability distribution used to simulate the data among "normal", "cauchy", "dexp", "student".
max_iter	integer, maximum number of iteration for the iterative simulation process.
verbose	boolean, if TRUE, enable verbosity.

Details

The c_val input argument should be strictly positive so that the null hypothesis is not verified when simulating the data (i.e. so that the two sample are not generated from the same probability distribution).

Value

a list with the following elements:

- power_res: a list of named numeric value corresponding to the power values for each statistic listed in stat input.
- pval_res: a list of named numeric values corresponding to the p-values for each simulation and each statistic listed in the stat input.
- simu_config: information about input parameters used for simulation, including n_simu, c_val, distrib, delta_shape, n_point, n_obs1, n_obs2.

References

Zaineb Smida, Lionel Cucala, Ali Gannoun & Ghislain Durif (2022) A median test for functional data, Journal of Nonparametric Statistics, 34:2, 520-553, doi:10.1080/10485252.2022.2064997, hal-03658578

See Also

stat_mo(), stat_med(), stat_wmw(), stat_hkr(), stat_cff(), comp_stat()

Examples

```
# simulate a few small data for the example
res <- power_exp(
    n_simu = 20, alpha = 0.05, n_perm = 100,
    stat = c("mo", "med", "wmw", "hkr", "cff"),
    n_point = 25, n_obs1 = 4, n_obs2 = 5, c_val = 10, delta_shape = "constant",
    distrib = "normal", max_iter = 10000, verbose = FALSE
)
res$power_res
```

`simul_data` *Simulation of trajectories from two samples diverging by a delta function*

Description

Simulate `n_obs1` trajectories of length `n_point` in the first sample and `n_obs2` trajectories of length `n_point` in the second sample.

Usage

```
simul_data(
  n_point,
  n_obs1,
  n_obs2,
  c_val = 0,
  delta_shape = "constant",
  distrib = "normal",
  max_iter = 10000
)
```

Arguments

`n_point`	integer value, number of points in the trajectory.
`n_obs1`	integer value, number of trajectories in the first sample.
`n_obs2`	integer value, number of trajectories in the second sample.
`c_val`	numeric value, level of divergence between the two samples.
`delta_shape`	character string, shape of the divergence between the two samples, among `"constant"`, `"linear"`, `"quadratic"`.
`distrib`	character string, type of probability distribution used to simulate the data among `"normal"`, `"cauchy"`, `"dexp"`, `"student"`.
`max_iter`	integer, maximum number of iteration for the iterative simulation process.

Value

A list with the following elements

- `mat_sample1`: numeric matrix of dimension `n_point x n_obs1` containing `n_obs1` trajectories (in columns) of size `n_point` (in rows) corresponding to sample 1.
- `mat_sample2`: numeric matrix of dimension `n_point x n_obs2` containing `n_obs2` trajectories (in columns) of size `n_point` (in rows) corresponding to sample 2.

References

Zaineb Smida, Lionel Cucala, Ali Gannoun & Ghislain Durif (2022) A median test for functional data, Journal of Nonparametric Statistics, 34:2, 520-553, doi:10.1080/10485252.2022.2064997, hal-03658578

See Also

`plot_simu()`, `simul_traj()`

Examples

```
simu_data <- simul_data(
    n_point = 100, n_obs1 = 50, n_obs2 = 75, c_val = 10,
    delta_shape = "constant", distrib = "normal"
)
str(simu_data)
```

---

simul_traj *Single trajectory simulation*

Description

Simulate a trajectory of length `n_point` using a random generator associated to different probability distribution.

Usage

```
simul_traj(n_point, distrib = "normal", max_iter = 10000)
```

Arguments

`n_point`	integer value, number of points in the trajectory.
`distrib`	character string, type of probability distribution used to simulate the data among `"normal"`, `"cauchy"`, `"dexp"`, `"student"`.
`max_iter`	integer, maximum number of iteration for the iterative simulation process.

Value

Vector of size `n_point` with the trajectory values.

References

Zaineb Smida, Lionel Cucala, Ali Gannoun & Ghislain Durif (2022) A median test for functional data, Journal of Nonparametric Statistics, 34:2, 520-553, doi:10.1080/10485252.2022.2064997, hal-03658578

See Also

simul_data()

Examples

```
simu_vec <- simul_traj(100)
plot(simu_vec, xlab = "point", ylab = "value")
```

---

stat_cff *Cuevas-Febrero-Fraiman statistic*

Description

The Cuevas-Febrero-Fraiman statistics defined in Cuevas et al (2004) (and noted CFF in Smida et al 2022) is computed to compare two sets of functional trajectories.

Usage

```
stat_cff(MatX, MatY)
```

Arguments

MatX	numeric matrix of dimension n_point x n containing n trajectories (in columns) of size n_point (in rows).
MatY	numeric matrix of dimension n_point x m containing m trajectories (in columns) of size n_point (in rows).

Value

numeric value corresponding to the WMW statistic value

References

Cuevas, A, Febrero, M, and Fraiman, R (2004) An anova test for functional data. Computational Statistics & Data Analysis, 47(1): 111–122. doi:10.1016/j.csda.2003.10.021

Zaineb Smida, Lionel Cucala, Ali Gannoun & Ghislain Durif (2022) A median test for functional data, Journal of Nonparametric Statistics, 34:2, 520-553, doi:10.1080/10485252.2022.2064997, hal-03658578

See Also

comp_stat(), permut_pval()

Examples

```
simu_data <- simul_data(
    n_point = 100, n_obs1 = 50, n_obs2 = 75, c_val = 10,
    delta_shape = "constant", distrib = "normal"
)

MatX <- simu_data$mat_sample1
MatY <- simu_data$mat_sample2

stat_cff(MatX, MatY)
```

`stat_hkr` *Horváth-Kokoszka-Reeder statistics*

Description

The Horváth-Kokoszka-Reeder statistics defined in Chakraborty & Chaudhuri (2015) (and noted HKR1 and HKR2 in Smida et al 2022) are computed to compare two sets of functional trajectories.

Usage

```
stat_hkr(MatX, MatY)
```

Arguments

`MatX`	numeric matrix of dimension `n_point x n` containing `n` trajectories (in columns) of size `n_point` (in rows).
`MatY`	numeric matrix of dimension `n_point x m` containing `m` trajectories (in columns) of size `n_point` (in rows).

Value

A list with the following elements

- `T1`: numeric value corresponding to the HKR1 statistic value
- `T2`: numeric value corresponding to the HKR2 statistic value
- `eigenval`: numeric vector of eigen values from the empirical pooled covariance matrix of `MatX` and `MatY` (see Smida et al, 2022, for more details)

References

Horváth, L., Kokoszka, P., & Reeder, R. (2013). Estimation of the mean of functional time series and a two-sample problem. Journal of the Royal Statistical Society. Series B (Statistical Methodology), 75(1), 103–122. doi:10.1111/j.14679868.2012.01032.x

Zaineb Smida, Lionel Cucala, Ali Gannoun & Ghislain Durif (2022) A median test for functional data, Journal of Nonparametric Statistics, 34:2, 520-553, doi:10.1080/10485252.2022.2064997, hal-03658578

See Also

comp_stat(), permut_pval()

Examples

```
simu_data <- simul_data(
    n_point = 100, n_obs1 = 50, n_obs2 = 75, c_val = 10,
    delta_shape = "constant", distrib = "normal"
)

MatX <- simu_data$mat_sample1
MatY <- simu_data$mat_sample2

stat_hkr(MatX, MatY)
```

---

stat_med *MED median statistic*

Description

The MED median statistics defined in Smida et al (2022) is computed to compare two sets of functional trajectories.

Usage

```
stat_med(MatX, MatY)
```

Arguments

MatX	numeric matrix of dimension n_point x n containing n trajectories (in columns) of size n_point (in rows).
MatY	numeric matrix of dimension n_point x m containing m trajectories (in columns) of size n_point (in rows).

Value

numeric value corresponding to the MED median statistic value

References

Zaineb Smida, Lionel Cucala, Ali Gannoun & Ghislain Durif (2022) A median test for functional data, Journal of Nonparametric Statistics, 34:2, 520-553, doi:10.1080/10485252.2022.2064997, hal-03658578

See Also

comp_stat(), permut_pval()

Examples

```
simu_data <- simul_data(
    n_point = 100, n_obs1 = 50, n_obs2 = 75, c_val = 10,
    delta_shape = "constant", distrib = "normal"
)

MatX <- simu_data$mat_sample1
MatY <- simu_data$mat_sample2

stat_med(MatX, MatY)
```

---

stat_mo *MO median statistic*

Description

The MO median statistics defined in Smida et al (2022) is computed to compare two sets of functional trajectories.

Usage

```
stat_mo(MatX, MatY)
```

Arguments

MatX	numeric matrix of dimension n_point x n containing n trajectories (in columns) of size n_point (in rows).
MatY	numeric matrix of dimension n_point x m containing m trajectories (in columns) of size n_point (in rows).

Value

numeric value corresponding to the MO median statistic value

References

Zaineb Smida, Lionel Cucala, Ali Gannoun & Ghislain Durif (2022) A median test for functional data, Journal of Nonparametric Statistics, 34:2, 520-553, doi:10.1080/10485252.2022.2064997, hal-03658578

See Also

comp_stat(), permut_pval()

Examples

```
simu_data <- simul_data(
    n_point = 100, n_obs1 = 50, n_obs2 = 75, c_val = 10,
    delta_shape = "constant", distrib = "normal"
)

MatX <- simu_data$mat_sample1
MatY <- simu_data$mat_sample2

stat_mo(MatX, MatY)
```

---

stat_wmw *Wilcoxon-Mann-Whitney (WMW) statistic*

Description

The Wilcoxon-Mann-Whitney statistic defined in Chakraborty & Chaudhuri (2015) (and noted WMW in Smida et al 2022) is computed to compare two sets of functional trajectories.

Usage

```
stat_wmw(MatX, MatY)
```

Arguments

`MatX`	numeric matrix of dimension `n_point x n` containing `n` trajectories (in columns) of size `n_point` (in rows).
`MatY`	numeric matrix of dimension `n_point x m` containing `m` trajectories (in columns) of size `n_point` (in rows).

Value

numeric value corresponding to the WMW statistic value

References

Anirvan Chakraborty, Probal Chaudhuri, A Wilcoxon–Mann–Whitney-type test for infinite-dimensional data, Biometrika, Volume 102, Issue 1, March 2015, Pages 239–246, doi:10.1093/biomet/asu072

Zaineb Smida, Lionel Cucala, Ali Gannoun & Ghislain Durif (2022) A median test for functional data, Journal of Nonparametric Statistics, 34:2, 520-553, doi:10.1080/10485252.2022.2064997, hal-03658578

See Also

`comp_stat()`, `permut_pval()`

Examples

```
simu_data <- simul_data(
    n_point = 100, n_obs1 = 50, n_obs2 = 75, c_val = 10,
    delta_shape = "constant", distrib = "normal"
)

MatX <- simu_data$mat_sample1
MatY <- simu_data$mat_sample2

stat_wmw(MatX, MatY)
```

Index

comp_stat, 2
comp_stat(), *4*, *6*, *9*, *11–13*

permut_pval, 3
permut_pval(), *9*, *11–13*
plot_simu, 4
plot_simu(), *8*
power_exp, 5

simul_data, 7
simul_data(), *4*, *9*
simul_traj, 8
simul_traj(), *8*
stat_cff, 9
stat_cff(), *3, 4*, *6*
stat_hkr, 10
stat_hkr(), *2–4*, *6*
stat_med, 11
stat_med(), *3, 4*, *6*
stat_mo, 12
stat_mo(), *3, 4*, *6*
stat_wmw, 13
stat_wmw(), *3, 4*, *6*