Package 'MD2sample'

November 2, 2026

Title Various Methods for the Two Sample Problem in D>1 Dimensions

Version 1.5.0

Description The routine twosample_test() in this package runs the
two-sample test using various test statistic for multivariate data.
The user can also run several tests and then find a p value adjusted for simultaneous inference.
The p values are found via permutation or via the parametric bootstrap.
The routine twosample_power() allows the estimation of the power of the tests.
The routine run.studies() allows a user to quickly study the power of a new method and
how it compares to those included in the package.
For details of the methods and references see the included vignettes.

License GPL (>= 2)

Encoding UTF-8

LinkingTo Rcpp

Imports Rcpp, parallel, stats, FNN, copula, ade4, gTests, igraph, lsa,
microbenchmark, mvtnorm, Ball

Suggests rmarkdown, knitr, ggplot2, egg, testthat (>= 3.0.0)

VignetteBuilder knitr

Depends R (>= 3.5)

LazyData true

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation yes

Author Wolfgang Rolke [aut, cre] (ORCID:
<https://orcid.org/0000-0002-3514-726X>)

Maintainer Wolfgang Rolke <wolfgang.rolke@upr.edu>

Repository CRAN

Date/Publication 2026-11-02 20:10:23 UTC

Contents

case.studies . . . 2
chisq2D_test_cont . . . 3
chisq2D_test_disc . . . 4
chiTS.cont . . . 4
chiTS.disc . . . 5
edge.tests . . . 5
FR.test . . . 6
MD2sample_power-methods . . . 7
MD2sample_test-methods . . . 8
method_info . . . 8
power_studies_results . . . 9
run.studies . . . 10
TS_cont_pval . . . 11
twosample_power . . . 12
twosample_test . . . 14
twosample_test_adjusted_pvalue . . . 16

Index **18**

`case.studies` *Create case studies*

Description

This function creates the functions needed to run the various case studies.

Usage

```
case.studies(
  which,
  n = 200,
  nx = n,
  ny = n,
  nbins = -1,
  Ranges = matrix(c(-Inf, Inf, -Inf, Inf), 2, 2),
  ReturnCaseNames = FALSE
)
```

Arguments

`which`	name of the case study.
`n`	=200, sample size for both data sets
`nx`	=n, sample size for first data sets
`ny`	=n, sample size for second data sets
`nbins`	=-1, number of bins for chi-square test, 2D only

Ranges	= matrix(c(-Inf, Inf, -Inf, Inf), 2, 2), ranges of variables
ReturnCaseNames	=FALSE, should list of case studies be returned?

Value

a list of functions and vectors

`chisq2D_test_cont` *This function does the chi square two sample test for continuous data in two dimensions*

Description

This function does the chi square two sample test for continuous data in two dimensions

Usage

```
chisq2D_test_cont(
  dta_x,
  dta_y,
  Ranges = matrix(c(-Inf, Inf, -Inf, Inf), 2, 2),
  nbins = c(5, 5),
  minexpcount = 5,
  SuppressMessages = FALSE
)
```

Arguments

dta_x	a matrix of numbers
dta_y	a matrix of numbers
Ranges	=matrix(c(-Inf, Inf, -Inf, Inf),2,2) a 2x2 matrix with lower and upper bounds
nbins	=c(5,5) number of bins in x and y direction
minexpcount	=5 minimum counts required per bin
SuppressMessages	=FALSE, print informative messages?

Value

a list with statistics and p values

chisq2D_test_disc — *This function does the chi square two sample test for continuous data in two dimensions*

Description

This function does the chi square two sample test for continuous data in two dimensions

Usage

```
chisq2D_test_disc(dta, minexpcount = 5)
```

Arguments

dta	a list with values and counts
minexpcount	=5 minimum counts required per bin

Value

a list with statistics and p values

chiTS.cont — *Chi-square test for continuous data, example of user supplied function*

Description

This function does the chi square two sample test for continuous data in two dimensions

Usage

```
chiTS.cont(x, y, TSextra)
```

Arguments

x	a matrix of numbers
y	a matrix of numbers
TSextra	list with some info

Value

either a test statistic or a p value

chiTS.disc *Chi-square test for discrete data, example of user supplied function*

Description

This function does the chi square two sample test for discrete data in two dimensions

Usage

```
chiTS.disc(x, y, vals_x, vals_y, TSextra)
```

Arguments

x	a vector with counts
y	a vector with counts
vals_x	a vector with points
vals_y	a vector with points
TSextra	list with some info

Value

either a test statistic or a p value

edge.tests *This function does the tests by Chen and Friedman*

Description

This function does the tests by Chen and Friedman

Usage

```
edge.tests(x, y)
```

Arguments

x	a matrix of numbers
y	a matrix of numbers

Value

a list with statistics and p values

FR.test *Friedman-Rafsky (FR) test*

Description

FR test is a multivariate generalization of nonparametric two-sample test. This function is an implementation with customized options, including a visualization of the minimum spanning tree (MST).

Usage

```
FR.test(
  samp1,
  samp2,
  use.cosine = FALSE,
  binary = FALSE,
  binary.cutoff = 2,
  plot.MST = FALSE,
  col = c("#F0E442", "#56B4E9"),
  label.names = c("Sample 1", "Sample 2"),
  vertex.size = 5,
  edge.width = 1,
  ...
)
```

Arguments

samp1	Numeric matrix or data frame for Sample 1. Rows are multivariate dimensions, and columns are samples. E.g. gene by cell.
samp2	Numeric matrix or data frame for Sample 2.
use.cosine	An option if to use cosine distance. Logical variable. By default (FALSE), Euclidean distance is used.
binary	An option if to use binary values. Logical variable. Default: FALSE. If TRUE, use binary.cutoff to dichotomize samp1 and samp2.
binary.cutoff	Numeric value for binary cutoff. Binary value = 1 if greater than binary.cutoff, 0 otherwise. Default: 2.
plot.MST	Boolean variable indicating if to plot the minimum spanning tree (MST). Default: FALSE.
col	Character vector of length two for customized colors of the nodes in MST. Default: c("#F0E442", "#56B4E9").
label.names	Character vector of length two for customized names of the two samples. Default: c("Sample 1","Sample 2").
vertex.size, edge.width, ...	Additional plotting parammeters passed to plot.igraph. Default: vertex.size=5, edge.width=1.

Value

Test statistics and p-values.

`runs`	Total number of subtrees.
`runs.samp1`	Number of subtrees of Sample 1.
`runs.samp2`	Number of subtrees of Sample 2.
`stat`	The standardized FR statistic.
`p.value`	P-value of the FR test.

`MD2sample_power-methods`

Methods for MD2sample power results

Description

Print and convert power estimates with Monte Carlo uncertainty.

Usage

```
## S3 method for class 'MD2sample_power'
print(x, ...)

## S3 method for class 'MD2sample_power'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	An object of class `"MD2sample_power"`.
`...`	Further arguments passed to the underlying print or data-frame method.
`row.names`	Ignored; included for compatibility with `as.data.frame`.
`optional`	Ignored; included for compatibility with `as.data.frame`.

Value

The print method returns `x` invisibly. The data-frame method returns the alternative parameter, method, power estimate, Monte Carlo standard error, and Wilson confidence limits.

MD2sample_test-methods

Methods for MD2sample two-sample test results

Description

Print, summarize, and convert structured `MD2sample_test` objects.

Usage

```
## S3 method for class 'MD2sample_test'
print(x, ...)

## S3 method for class 'MD2sample_test'
summary(object, ...)

## S3 method for class 'summary.MD2sample_test'
print(x, ...)

## S3 method for class 'MD2sample_test'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	An object of class `"MD2sample_test"`, or for `print.summary.MD2sample_test`, an object of class `"summary.MD2sample_test"`.
`...`	Further arguments passed to the relevant method.
`object`	An object of class `"MD2sample_test"`.
`row.names`	Ignored; included for compatibility with `as.data.frame`.
`optional`	Ignored; included for compatibility with `as.data.frame`.

Value

The print methods return their input invisibly; `summary()` returns a summary object; `as.data.frame()` returns columns `method`, `statistic`, and `p.value`.

method_info

Information about the included methods

Description

Returns a table with one row for each method included in MD2sample: the code used in the output of `twosample_test()` and `twosample_power()`, its full name, the type of data, how p values are found, a reference, and notes on methods that are identical to others.

Usage

```
method_info(Continuous = NULL)
```

Arguments

Continuous	=NULL, if TRUE or FALSE only the methods for continuous or discrete data are returned.

Details

Some methods are the same test under different names: FR and CF1 are both the original edge-count test of Friedman and Rafsky on the minimum spanning tree, and CF3, the weighted edge-count test of Chen and Friedman, is identical to CF1 if the two data sets have the same sample size. The codes CF1 to CF4 follow the order in which `gTests::g.tests` returns the tests: original, generalized, weighted and max-type edge-count test.

Value

A data frame with columns `code`, `data`, `name`, `p.value`, `reference` and `note`.

Examples

```
method_info(Continuous=TRUE)
```

`power_studies_results` *power_studies_results*

Description

the results of the included power studies

Usage

```
power_studies_results
```

Format

'power_studies_results':

A list of matrices with powers

run.studies *Benchmarking for Multivariate Two-Sample Tests*

Description

This function runs the case studies included in the package.

Usage

```
run.studies(
  study,
  Continuous = TRUE,
  TS,
  TSextra,
  With.p.value = FALSE,
  alpha = 0.05,
  param_alt,
  nsample = c(200, 200),
  SuppressMessages = FALSE,
  B = 1000,
  maxProcessor,
  seed = NULL
)
```

Arguments

study	either the name of the study, or its number in the list. If missing all the studies are run.
Continuous	=TRUE, run cases for continuous data.
TS	routine to calculate new test statistics.
TSextra	list passed to TS (optional).
With.p.value	=FALSE, does user supplied routine return p values?
alpha	=0.05, type I error probability of tests. 0.05 is used in included case studies.
param_alt	vector or matrix of values of parameters under the alternative hypothesis. If missing included values are used.
nsample	=c(200, 200), sample sizes for x and y data sets.
SuppressMessages	=FALSE, should informative messages be printed?
B	= 1000, number of simulation runs.
maxProcessor	number of cores to use. If missing the number of physical cores-1 is used. If set to 1 no parallel processing is done.
seed	=NULL, optional random-number seed. Case study number k in the list returned by case.studies(ReturnCaseNames=TRUE) uses the seed seed+k, so any single case study can be reproduced on its own.

Details

For details consult vignette(package="MD2sample")

Value

A (list of) matrices of p.values.

Examples

```
#The new test is a (included) chi square test:
TSextra=list(which="pval", nbins=rbind(c(3,3), c(4,4)))
run.studies(study=c("NormalD2", "tD2"), Continuous=TRUE,
          TS=chiTS.cont, TSextra=TSextra,
          With.p.value=TRUE, B=100)
```

TS_cont_pval *This function runs a number of two sample tests which find their own p value.*

Description

This function runs a number of two sample tests which find their own p value.

Usage

```
TS_cont_pval(x, y, DoBall = TRUE)
```

Arguments

x	a matrix of numbers if data is continuous or of counts if data is discrete.
y	a matrix of numbers if data is continuous or of counts if data is discrete.
DoBall	=TRUE, run the Ball divergence test? Its large sample p value needs the eigenvalues of a matrix of the size of the combined data set, so it is slow for large data sets. If FALSE its statistic and p value are NA.

Value

A list of two numeric vectors, the test statistics and the p values.

`twosample_power` *Power Estimation for Multivariate Two-Sample Tests*

Description

Estimate the power of various two sample tests using Rcpp and parallel computing.

Usage

```
twosample_power(
  f,
  ...,
  TS,
  TSextra,
  alpha = 0.05,
  B = 1000,
  nbins = c(5, 5),
  minexpcount = 5,
  Ranges = matrix(c(-Inf, Inf, -Inf, Inf), 2, 2),
  samplingmethod = "Binomial",
  rnull,
  With.p.value = FALSE,
  DoTransform = TRUE,
  SuppressMessages = FALSE,
  LargeSampleOnly = FALSE,
  maxProcessor,
  doMethods = "all",
  CI = FALSE,
  conf.level = 0.95,
  seed = NULL
)
```

Arguments

`f`	function to generate a list with data sets x and y for continuous data or a four-column numeric matrix for discrete data. Standard names vals_x, vals_y, x and y are used when present; otherwise the two support/bin and two count columns are inferred from their numbers of distinct values.
`...`	additional arguments passed to f, up to 2.
`TS`	routine to calculate test statistics for new tests. For discrete data it may have signature `TS(x)` or `TS(x, TSextra)`, where `x` is the four-column discrete-data matrix; the legacy 4- or 5-argument form is also accepted.
`TSextra`	additional info passed to TS, if necessary.
`alpha`	=0.05, the type I error probability of the hypothesis test.
`B`	=1000, number of simulation runs.

nbins	=c(5, 5), number of bins for chi square test if Dim=2.
minexpcount	=5, lowest required count for chi-square test.
Ranges	=matrix(c(-Inf, Inf, -Inf, Inf),2,2), a 2x2 matrix with lower and upper bounds.
samplingmethod	="Binomial" for Binomial sampling or "independence" for independence sampling in the discrete data case.
rnull	function to generate new data sets for parametric bootstrap.
With.p.value	=FALSE, does user supplied routine return p values?
DoTransform	=TRUE, should data be transformed to to unit hypercube?
SuppressMessages	=FALSE, should messages be printed?
LargeSampleOnly	=FALSE, should only methods with large sample theories be run?
maxProcessor	number of cores to use. If missing the number of physical cores-1 is used. If set to 1 no parallel processing is done.
doMethods	="all", which methods should be included?
CI	=FALSE if TRUE, return Monte Carlo standard errors and Wilson confidence intervals.
conf.level	=0.95 confidence level used when CI=TRUE.
seed	=NULL optional random-number seed for reproducibility.

Details

For details consult vignette("MD2sample","MD2sample")

Value

By default, a numeric matrix or vector of power values. If CI=TRUE, an object of class "MD2sample_power" with power estimates, Monte Carlo standard errors, and Wilson confidence limits.

Examples

```
#Note that the resulting power estimates are meaningless because
#of the extremely low number of simulation runs B, required because of CRAN timing rule
#
#Power of tests when one data set comes from a standard normal multivariate distribution function
#and the other data set from a multivariate normal with correlation
#number of simulation runs is ridiculously small because of CRAN submission rules
f=function(a=0) {
 S=diag(2)
 x=mvtnorm::rmvnorm(100, sigma = S)
 S[1,2]=a
 S[2,1]=a
 y=mvtnorm::rmvnorm(120, sigma = S)
 list(x=x, y=y)
}
twosample_power(f, c(0, 0.5), B=10, maxProcessor=1)
#Power of use supplied test. Example is a (included) chi-square test:
```

```
TSextra=list(which="statistics", nbins=rbind(c(3,3), c(4,4)))
twosample_power(f, c(0, 0.5), TS=chiTS.cont, TSextra=TSextra, B=10, maxProcessor=1)
#Same example, but this time the user supplied routine calculates p values:
TSextra=list(which="pvalues", nbins=c(4,4))
twosample_power(f, c(0, 0.5), TS=chiTS.cont, TSextra=TSextra, B=10,
            With.p.value=TRUE, maxProcessor=1)
#Example for discrete data
g=function(p1, p2) {
  x = table(sample(1:4, size=1000, replace = TRUE))
  y = table(sample(1:4, size=500, replace = TRUE, prob=c(p1,p2,1,1)))
  cbind(vals_x=rep(1:2,2),  vals_y=rep(1:2, each=2), x=x, y=y)
}
twosample_power(g, 1.5, 1.6, B=10, maxProcessor=1)
```

`twosample_test` *Multivariate Two-Sample Tests*

Description

This function runs a number of two sample tests using Rcpp and parallel computing.

Usage

```
twosample_test(
  x,
  y,
  vals_x = NA,
  vals_y = NA,
  TS,
  TSextra,
  B = 5000,
  seed = NULL,
  nbins = c(5, 5),
  minexpcount = 5,
  Ranges = matrix(c(-Inf, Inf, -Inf, Inf), 2, 2),
  DoTransform = TRUE,
  samplingmethod = "Binomial",
  rnull,
  SuppressMessages = FALSE,
  LargeSampleOnly = FALSE,
  maxProcessor,
  doMethods = "all"
)
```

Arguments

x	Continuous data: either a matrix of numbers, or a list with two matrices called x and y. if it is a matrix Observations are in different rows. Discrete data: a vector

	of counts or a four-column numeric matrix containing two support/bin columns and two count columns. Standard names vals_x, vals_y, x and y are used when present; otherwise the roles are inferred from the numbers of distinct values.
y	a matrix of numbers if data is continuous or a vector of counts if data is discrete.
vals_x	=NA, a vector of values for discrete random variables, or NA if data is continuous.
vals_y	=NA, a vector of values for discrete random variables, or NA if data is continuous.
TS	user supplied routine to calculate test statistics for new tests. For discrete data it may have signature TS(x) or TS(x, TSextra), where x is the four-column discrete-data matrix; the legacy 4- or 5-argument form is also accepted.
TSextra	(optional) additional info passed to TS, if necessary.
B	=5000, number of simulation runs for permutation test.
seed	=NULL, optional seed for simulations
nbins	=c(5,5), for chi square tests (2D only).
minexpcount	=5, lowest required count for chi-square test (2D only).
Ranges	=matrix(c(-Inf, Inf, -Inf, Inf),2,2), a 2x2 matrix with lower and upper bounds (2D only).
DoTransform	=TRUE, should data be transformed to unit hypercube?
samplingmethod	="Binomial" for Binomial sampling or "independence" for independence sampling.
rnull	function to generate new data sets for simulation as an alternative to the permutation method.
SuppressMessages	=FALSE, should informative messages be printed?
LargeSampleOnly	=FALSE, should only methods with large sample theories be run?
maxProcessor	number of cores to use. If missing the number of physical cores-1 is used. If set to 1 no parallel processing is done.
doMethods	="all", Which methods should be included?

Details

For details consult vignette("MD2sample","MD2sample")

Value

An object of class "MD2sample_test" with components results, statistics, p.values, metadata, and call. The traditional $statistics and $p.values components are retained for backward compatibility.

Examples

```
#Two continuous data sets from a multivariate normal:
x = mvtnorm::rmvnorm(100, c(0,0))
y = mvtnorm::rmvnorm(120, c(0,0))
twosample_test(x, y, B=100, maxProcessor=1)
#Using a new test, this one is an (included) chi square test.
#Also enter data as a list:
TSextra=list(which="statistics", nbins=rbind(c(3,3), c(4,4)))
dta=list(x=x, y=y)
twosample_test(dta, TS=chiTS.cont, TSextra=TSextra, B=100, maxProcessor=1)
#Two discrete data sets from some distribution:
x = table(sample(1:4, size=1000, replace = TRUE))
y = table(sample(1:4, size=1000, replace = TRUE, prob=c(1,2,1,1)))
vals_x=rep(1:2,2)
vals_y=rep(1:2, each=2)
twosample_test(x, y, vals_x, vals_y, B=100, maxProcessor=1)
#Run a discrete chi square test and enter the data as a matrix:
TSextra=list(which="statistics")
dta=cbind(x=x, y=y, vals_x=vals_x, vals_y=vals_y)
twosample_test(dta, TS=chiTS.disc, TSextra=TSextra, B=100, maxProcessor=1)
```

---

twosample_test_adjusted_pvalue

Adjusted p values for multivariate two-sample tests

Description

Runs several two-sample tests and returns their individual p-values together with a simulation-based p-value for the minimum p-value across the selected tests.

Usage

```
twosample_test_adjusted_pvalue(
  x,
  y,
  vals_x = NA,
  vals_y = NA,
  B = c(5000, 1000),
  nbins = c(5, 5),
  minexpcount = 5,
  samplingmethod = "Binomial",
  Ranges = matrix(c(-Inf, Inf, -Inf, Inf), 2, 2),
  DoTransform = TRUE,
  rnull,
  SuppressMessages = FALSE,
  maxProcessor,
  doMethods,
  seed = NULL
)
```

Arguments

x, y, vals_x, vals_y	Data arguments as in twosample_test().
B	Length-one or length-two vector giving simulation sizes.
nbins, minexpcount, Ranges	Chi-square settings.
samplingmethod	Sampling method for discrete data.
DoTransform	Should continuous data be transformed to the unit hypercube?
rnull	Optional parametric-bootstrap generator.
SuppressMessages	Suppress informative messages?
maxProcessor	Number of processors.
doMethods	Methods to combine. If missing, a default subset is used.
seed	Optional random-number seed for reproducibility.

Value

A named numeric vector containing the individual p-values and the adjusted minimum-p value in the final element, named "Min p".

Index

∗ **datasets**
power_studies_results, 9

as.data.frame.MD2sample_power
(MD2sample_power-methods), 7
as.data.frame.MD2sample_test
(MD2sample_test-methods), 8

case.studies, 2
chisq2D_test_cont, 3
chisq2D_test_disc, 4
chiTS.cont, 4
chiTS.disc, 5

edge.tests, 5

FR.test, 6

MD2sample_power-methods, 7
MD2sample_test-methods, 8
method_info, 8

plot.igraph, *6*
power_studies_results, 9
print.MD2sample_power
(MD2sample_power-methods), 7
print.MD2sample_test
(MD2sample_test-methods), 8
print.summary.MD2sample_test
(MD2sample_test-methods), 8

run.studies, 10

summary.MD2sample_test
(MD2sample_test-methods), 8

TS_cont_pval, 11
twosample_power, 12
twosample_test, 14
twosample_test_adjusted_pvalue, 16