Package: ICIKendallTau (via r-universe)

August 21, 2024

Title Calculates information-content-informed Kendall-tau

Version 1.2.1

Description Provides functions for calculating information-content-informed Kendall-tau. This version of Kendall-tau allows for the inclusion of missing values.

VignetteBuilder knitr

LazyData true

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

LinkingTo Rcpp

Imports Rcpp, purrr, utils, stringr, stats, rlang, cli

Suggests furrr, future, testthat (>= 3.0.0), microbenchmark, rmarkdown, knitr, dplyr, logger, withr, ggplot2, visdat

URL https://moseleybioinformaticslab.github.io/ICIKendallTau/ https://github.com/moseleybioinformaticslab/ICIKendallTau/

BugReports https://github.com/moseleybioinformaticslab/ICIKendallTau/issues

Config/testthat/edition 3

Depends R (>= 3.5)

Repository https://moseleybioinformaticslab.r-universe.dev

RemoteUrl https://github.com/MoseleyBioinformaticsLab/ICIKendallTau

RemoteRef v_1.2.1

RemoteSha 07dcab7333f19c6843ca500463d8a214142183d9

Contents

add_uniform_noise . . . 2
calculate_matrix_medians . . . 3
cor_fast . . . 3
cor_matrix_2_long_df . . . 4
disable_logging . . . 5
enable_logging . . . 5
ici_kendalltau . . . 6
ici_kt . . . 8
kt_fast . . . 9
log_memory . . . 11
log_message . . . 11
long_df_2_cor_matrix . . . 11
missing_dataset . . . 12
pairwise_completeness . . . 12
rank_order_data . . . 13
show_progress . . . 13
test_left_censorship . . . 14
yeast_missing . . . 15

Index **16**

`add_uniform_noise` *Add uniform noise*

Description

Adds uniform noise to values, generating replicates with noise added to the original.

Usage

```
add_uniform_noise(value, n_rep, sd, use_zero = FALSE)
```

Arguments

`value`	a single or vector of numeric values
`n_rep`	the number of replicates to make (numeric). Default is 1.
`sd`	the standard deviation of the data
`use_zero`	logical, should returned values be around zero or not?

Value

numeric matrix

calculate_matrix_medians *Calculate matrix medians*

Description

Given a matrix of data, calculates the median value in each column or row.

Usage

```
calculate_matrix_medians(in_matrix, use = "col", ...)
```

Arguments

in_matrix	numeric matrix of values
use	character of "col" or "row" defining columns or rows
...	extra parameters to the median function

Value

numeric

cor_fast *Fast correlation with test*

Description

Allows to run `cor.test` on a matrix of inputs.

Usage

```
cor_fast(
  x,
  y = NULL,
  use = "everything",
  method = "pearson",
  alternative = "two.sided",
  continuity = FALSE,
  return_matrix = TRUE
)
```

Arguments

x	a numeric vector, matrix, or data frame.
y	NULL (default) or a vector.
use	an optional character string giving a method for computing correlations in the presence of missing values. This must be (an abbreviation of) one of the strings "everything", "all.obs", "complete.obs", or "pairwise.complete.obs".
method	which correlation method to use, "pearson" or "spearman"
alternative	how to perform the statistical test
continuity	should a continuity correction be applied
return_matrix	should the matrices of values be returned, or a long data.frame

Details

Although the interface is *mostly* identical to the built-in `stats::cor.test()` method, there are some differences.

- if only `x` is provided as a matrix, the columns must be named.
- if providing both `x` and `y`, it is assumed they are both single vectors.
- if `NA` values are present, this function does not error, but will either remove them or return `NA`, depending on the option.
- "na.or.complete" is not a valid option for `use`.
- A named list with matrices or data.frame is returned, with the `rho` and `pvalue` values.

Value

a list of matrices, rho, pvalue, or a data.frame.

`cor_matrix_2_long_df` *convert matrix to data.frame*

Description

Given a square correlation matrix, converts it to a long data.frame, with three columns.

Usage

```
cor_matrix_2_long_df(in_matrix)
```

Arguments

in_matrix	the correlation matrix

Details

The data.frame contains three columns:

- s1: the first entry of comparison
- s2: the second entry of comparison
- cor: the correlation value

Value

data.frame

---

disable_logging *turn logging off*

Description

There may be good reasons to turn the logging off after it's been turned on. This basically tells the package that the logger isn't available.

Usage

```
disable_logging()
```

---

enable_logging *turn logging on*

Description

Choose to enable logging, to a specific file if desired.

Usage

```
enable_logging(log_file = NULL, memory = FALSE)
```

Arguments

log_file	the file to log to
memory	provide memory logging too? Only available on Linux and MacOS

Details

Uses the logger package under the hood, which is suggested in the dependencies. Having logging enabled is nice to see when things are starting and stopping, and what exactly has been done, without needing to write messages to the console. It is especially useful if you are getting errors, but can't really see them, then you can add "memory" logging to see if you are running out of memory.

Default log file has the pattern:

YYYY.MM.DD.HH.MM.SS_ICIKendallTau_run.log

`ici_kendalltau` *Information-content-informed kendall tau*

Description

Given a data-matrix, computes the information-theoretic Kendall-tau-b between all samples.

Usage

```
ici_kendalltau(
  data_matrix,
  global_na = c(NA, Inf, 0),
  perspective = "global",
  scale_max = TRUE,
  diag_good = TRUE,
  include_only = NULL,
  alternative = "two.sided",
  continuity = FALSE,
  check_timing = FALSE,
  return_matrix = TRUE
)
```

Arguments

`data_matrix`	matrix or data.frame of values, samples are columns, features are rows
`global_na`	numeric vector that defines globally, what should be treated as NA?
`perspective`	how to treat missing data in denominator and ties, character
`scale_max`	logical, should everything be scaled compared to the maximum correlation?
`diag_good`	logical, should the diagonal entries reflect how many entries in the sample were "good"?
`include_only`	only run the correlations that include the members (as a vector) or combinations (as a list or data.frame)
`alternative`	what is the alternative for the p-value test?
`continuity`	should a continuity correction be applied?
`check_timing`	logical to determine should we try to estimate run time for full dataset? (default is FALSE)
`return_matrix`	logical, should the data.frame or matrix result be returned?

Details

For more details, see the vignette `vignette("ici-kendalltau", package = "ICIKendallTau"))`

The default for `global_na` includes what values in the data to replace with NA for the Kendall-tau calculation. By default these are `global_na = c(NA, Inf, 0)`. If you want to replace something

other than 0, for example, you might use `global_na = c(NA, Inf, -2)`, and all values of -2 will be replaced instead of 0.

When `check_timing = TRUE`, 5 random pairwise comparisons will be run to generate timings on a single core, and then estimates of how long the full set will take are calculated. The data is returned as a data.frame, and will be on the low side, but it should provide you with a good idea of how long your data will take.

Returned is a list containing matrices with:

- cor: scaled correlations
- raw: raw kendall-tau correlations
- pval: p-values
- taumax: the theoretical maximum kendall-tau value possible

Eventually, we plan to provide two more parameters for replacing values, `feature_na` for feature specific NA values and `sample_na` for sample specific NA values.

If you want to know if the missing values in your data are possibly due to left-censorship, we recommend testing that hypothesis with `test_left_censorship()` first.

Value

list with cor, raw, pval, taumax

See Also

`test_left_censorship()` `pairwise_completeness()` `kt_fast()`

Examples

```
## Not run:
# not run
set.seed(1234)
s1 = sort(rnorm(1000, mean = 100, sd = 10))
s2 = s1 + 10

matrix_1 = cbind(s1, s2)

r_1 = ici_kendalltau(matrix_1)
r_1$cor

#    s1 s2
# s1  1  1
# s2  1  1
names(r_1)
# "cor", "raw", "pval", "taumax", "keep", "run_time"

s3 = s1
s3[sample(100, 50)] = NA

s4 = s2
s4[sample(100, 50)] = NA
```

```
matrix_2 = cbind(s3, s4)
r_2 = ici_kendalltau(matrix_2)
r_2$cor
#           s3        s4
# s3 1.0000000 0.9944616
# s4 0.9944616 1.0000000

# using include_only
set.seed(1234)
x = t(matrix(rnorm(5000), nrow = 100, ncol = 50))
colnames(x) = paste0("s", seq(1, nrow(x)))

# only calculate correlations of other columns with "s1"
include_s1 = "s1"
s1_only = ici_kendalltau(x, include_only = include_s1)

# include s1 and s3 things both
include_s1s3 = c("s1", "s3")
s1s3_only = ici_kendalltau(x, include_only = include_s1s3)

# only specify certain pairs either as a list
include_pairs = list(g1 = "s1", g2 = c("s2", "s3"))
s1_other = ici_kendalltau(x, include_only = include_pairs)

# or a data.frame
include_df = as.data.frame(list(g1 = "s1", g2 = c("s2", "s3")))
s1_df = ici_kendalltau(x, include_only = include_df)


## End(Not run)
```

`ici_kt` *Calculates ici-kendall-tau*

Description

Calculates ici-kendall-tau

Usage

```
ici_kt(
  x,
  y,
  perspective = "local",
  alternative = "two.sided",
  continuity = FALSE,
  output = "simple"
)
```

Arguments

x	numeric vector
y	numeric vector
perspective	should we consider the "local" or "global" perspective?
alternative	what is the alternative for the p-value test?
continuity	logical: if true, a continuity correction is used
output	used to control reporting of values for debugging

Details

Calculates the information-content-informed Kendall-tau correlation measure. This correlation is based on concordant and discordant ranked pairs, like Kendall-tau, but also includes missing values (as NA). Missing values are assumed to be *primarily* due to lack of detection due to instrumental sensitivity, and therefore encode *some* information.

For more details see the ICI-Kendall-tau vignette:

```
browseVignettes("ICIKendallTau")
```

Value

kendall tau correlation, p-value, max-correlation

Examples

```
x = sort(rnorm(100))
y = x + 1
y2 = y
y2[1:10] = NA
ici_kt(x, y)
ici_kt(x, y2, "global")
ici_kt(x, y2)
```

---

kt_fast *Fast kendall tau*

Description

Uses the underlying c++ implementation of `ici_kt` to provide a fast version of Kendall-tau correlation.

Usage

```
kt_fast(
  x,
  y = NULL,
  use = "everything",
  alternative = "two.sided",
  continuity = FALSE,
  return_matrix = TRUE
)
```

Arguments

x	a numeric vector, matrix, or data frame.
y	NULL (default) or a vector.
use	an optional character string giving a method for computing correlations in the presence of missing values. This must be (an abbreviation of) one of the strings "everything", "all.obs", "complete.obs", or "pairwise.complete.obs".
alternative	the type of test
continuity	should a continuity correction be applied
return_matrix	Should the matrices of values be returned, or a long data.frame

Details

Although the interface is *mostly* identical to the built-in `stats::cor()` method, there are some differences.

- if only `x` is provided as a matrix or data.frame, the columns must be named.
- if providing both `x` and `y`, it is assumed they are both single vectors.
- if `NA` values are present, this function does not error, but will either remove them or return `NA`, depending on the option.
- "na.or.complete" is not a valid option for `use`.
- A named list with matrices or data.frame is returned, with the `tau` and `pvalue` values.

Value

a list of matrices, tau, pvalue, or a data.frame.

log_memory *log memory usage*

Description

Logs the amount of memory being used to a log file if it is available, and generating warnings if the amount of RAM hits zero.

Usage

```
log_memory()
```

log_message *log messages*

Description

If a log_appender is available, logs the given message at the `info` level.

Usage

```
log_message(message_string)
```

Arguments

`message_string`	the string to put in the message

long_df_2_cor_matrix *convert data.frame to matrix*

Description

Given a long data.frame, converts it to a possibly square correlation matrix

Usage

```
long_df_2_cor_matrix(long_df, is_square = TRUE)
```

Arguments

`long_df`	the long data.frame
`is_square`	should it be a square matrix?

Value

matrix

missing_dataset *Example Dataset With Missingness*

Description

An example dataset that has missingness from left-censorship

Usage

```
missing_dataset
```

Format

`missing_dataset`**:**
A matrix with 1000 rows and 20 columns, where rows are features and columns are samples.

Source

Robert M Flight

pairwise_completeness *pairwise completeness*

Description

Calculates the completeness between any two samples using "or", is an entry missing in either X "or" Y.

Usage

```
pairwise_completeness(
  data_matrix,
  global_na = c(NA, Inf, 0),
  include_only = NULL,
  return_matrix = TRUE
)
```

Arguments

`data_matrix`	samples are columns, features are rows
`global_na`	globally, what should be treated as NA?
`include_only`	is there certain comparisons to do?
`return_matrix`	should the matrix or data.frame be returned?

Value

matrix of degree of completeness

See Also

ici_kendalltau()

`rank_order_data` *Rank order row data*

Description

Given a data-matrix of numeric data, calculates the rank of each row in each column (feature in sample), gets the median rank across all columns, and returns the original data with missing values set to NA, the reordered data, and a data.frame of the ranks of each feature and the number of missing values.

Usage

```
rank_order_data(data_matrix, global_na = c(NA, Inf, 0), sample_classes = NULL)
```

Arguments

`data_matrix`	matrix or data.frame of values
`global_na`	the values to consider as missing
`sample_classes`	are the columns defined by some metadata?

Value

list with two matrices and a data.frame

`show_progress` *turn progress on off*

Description

Allow the user to turn progress messages to the console and off. Default is to provide messages to the console.

Usage

```
show_progress(progress = TRUE)
```

Arguments

`progress`	logical to have it on or off

`test_left_censorship` *Test for left censorship*

Description

Does a binomial test to check if the most likely cause of missing values is due to values being below the limit of detection, or coming from a left-censored distribution.

Usage

```
test_left_censorship(
  data_matrix,
  global_na = c(NA, Inf, 0),
  sample_classes = NULL
)
```

Arguments

`data_matrix`	matrix or data.frame of numeric data
`global_na`	what represents zero or missing?
`sample_classes`	which samples are in which class

Details

For each feature that is missing in a group of samples, we save as a possibility to test. For each sample, we calculate the median value with any missing values removed. Each feature that had a missing value, we test whether the remaining non-missing values are below the sample median for those samples where the feature is non-missing. A binomial test considers the total number of features instances (minus missing values) as the number of trials, and the number of of features below the sample medians as the number of successes.

There is a bit more detail in the vignette: `vignette("testing-for-left-censorship", package = "ICIKendallTau")`

Value

data.frame of trials / successes, and binom.test result

See Also

`ici_kendalltau()`

Examples

```
# this example has 80% missing due to left-censorship
data(missing_dataset)
missingness = test_left_censorship(missing_dataset)
missingness$values
missingness$binomial_test
```

`yeast_missing` *Example RNA-Seq Dataset With Missingness*

Description

An example dataset from RNA-seq experiment on yeast, created by Gierliński et al., "Statistical models for RNA-seq data derived from a two-condition 48-replicate experiment", Bioinformatics, 31, 2015 https://doi.org/10.1093/bioinformatics/btv425.

Usage

```
yeast_missing
```

Format

`yeast_missing`**:**

A matrix with 6887 rows (genes) and 96 columns (samples).

Source

https://dx.doi.org/10.6084/M9.FIGSHARE.1425502.V1 https://dx.doi.org/10.6084/M9.FIGSHARE.1425503.V1

Index

∗ **datasets**
missing_dataset, 12
yeast_missing, 15

add_uniform_noise, 2

calculate_matrix_medians, 3
cor_fast, 3
cor_matrix_2_long_df, 4

disable_logging, 5

enable_logging, 5

ici_kendalltau, 6
ici_kendalltau(), *13, 14*
ici_kt, 8

kt_fast, 9
kt_fast(), *7*

log_memory, 11
log_message, 11
long_df_2_cor_matrix, 11

missing_dataset, 12

pairwise_completeness, 12
pairwise_completeness(), *7*

rank_order_data, 13

show_progress, 13
stats::cor(), *10*
stats::cor.test(), *4*

test_left_censorship, 14
test_left_censorship(), *7*

yeast_missing, 15