

Package ‘covercorr’

July 2, 2026

Title Coverage Correlation Coefficient and Testing for Independence

Version 1.1.0

Maintainer Tengyao Wang <t.wang59@lse.ac.uk>

Description Computes the coverage correlation coefficient introduced in <[doi:10.48550/arXiv.2508.06402](https://doi.org/10.48550/arXiv.2508.06402)> , a statistical measure that quantifies dependence between two random vectors by computing the union volume of data-centered hypercubes in a uniform space.

License GPL-3

Imports transport, MASS, fields, grDevices

Encoding UTF-8

RoxygenNote 7.3.3

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation yes

Author Tengyao Wang [aut, cre],
Mona Azadkia [aut, ctb],
Xuzhi Yang [aut, ctb]

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2026-07-02 16:20:02 UTC

Contents

CD8T	2
coverage_correlation	2
coverage_correlation_grid	4
coverage_correlation_K	6
coverage_correlation_K_grid	8
covercorr	9
covered_volume	11

covered_volume_mc	12
covered_volume_partitioned	13
MK_rank	13
plot_rectangles	14
split_rectangles	15
variance_formula	16
visualise_density	16
Index	18

CD8T	<i>Dataset: CD8+ T cell gene expression data</i>
------	--

Description

The CD8T dataset provides the gene expression data of fetal CD8+ T cells obtained in a single-cell RNA-seq experiment.

Usage

data(CD8T)

Format

A data frame with 9369 rows (cells) and 1000 columns (genes).

Source

Suo et al., Science (2022).

References

Suo, C., Dann, E., Goh, I., Jardine, L., Kleshchevnikov, V., Park, J.-E., Botting, R. A., et al. "Mapping the developing human immune system across organs." Science 376(6597), eabo0510 (2022).

coverage_correlation	<i>Coverage-based Dependence Measure with Optional Visualisation</i>
----------------------	--

Description

Computes the coverage correlation coefficient between input x and y, as introduced in [the arXiv preprint](#). This coefficient measures the dependence between two random variables or vectors.

Usage

```
coverage_correlation(
  x,
  y,
  visualise = FALSE,
  method = c("auto", "exact", "approx"),
  M = NULL,
  na.rm = TRUE
)
```

Arguments

x	Numeric vector or matrix.
y	Numeric vector or matrix with the same number of rows as x.
visualise	Logical; if TRUE, displays a scatter plot of the rank-transformed points with overlaid rectangles to illustrate the coverage calculation. The default is FALSE (no plot). If set to TRUE but either x or y has more than one column, a warning is issued and visualise is reset to FALSE.
method	Character string specifying the computation method. Options are "auto", "exact", or "approx". See Details.
M	Integer; Number of Monte Carlo integration sample points (used when method = "approx"). Optional.
na.rm	Logical; if TRUE, remove NA values before computation.

Details

The procedure is as follows:

1. Calculate the rank transformations (r_x, r_y) of the inputs x and y.
2. Construct small cubes (in 2D, squares) of volume n^{-1} centered at each rank-transformed point.
3. Compute the total area of the union of these cubes, intersected with $[0, 1]^d$ where $d = d_x + d_y$.

The coverage correlation coefficient is then calculated based on this union area.

For more details, please refer to the original paper: [the arXiv preprint](#).

The method argument controls how the computation is performed:

- "exact": Computes the exact value.
- "approx": Uses a Monte Carlo approximation with M sample points.
- "auto": Automatically selects a method based on the total number of columns in x and y: if more than 6, "approx" is used (with $M = \text{nrow}(x)^{\{1.5\}}$ if M is not provided); otherwise, "exact" is used.

Value

A list with four elements:

- `stat` – The numeric value of the coverage correlation coefficient.
- `pval` – The p-value, calculated using the exact variance under the null hypothesis of independence between `x` and `y`.
- `method` – A character string indicating the computation method used.
- `mc_se` – A numeric value. If method "approx" was used `mc_se` is the standard error of the Monte Carlo approximation, otherwise it is 0.

Examples

```
set.seed(1)
n <- 100
x <- runif(n)
y <- sin(3*x) + runif(n) * 0.01
coverage_correlation(x, y, visualise = TRUE)
```

coverage_correlation_grid

Coverage-based Dependence Measure with Optional Reference Grids

Description

Identical to [coverage_correlation](#), except that the reference points used for the Monge–Kantorovich rank transformation may be supplied directly via `u` and `v` rather than being generated internally from `runif`. This is useful for reproducibility, for using low-discrepancy (e.g. quasi-Monte Carlo) reference grids, or for comparing results across calls with a fixed reference set.

Usage

```
coverage_correlation_grid(
  x,
  y,
  u = NULL,
  v = NULL,
  visualise = FALSE,
  method = c("auto", "exact", "approx"),
  M = NULL,
  na.rm = TRUE
)
```

Arguments

<code>x</code>	Numeric vector or matrix.
<code>y</code>	Numeric vector or matrix with the same number of rows as <code>x</code> .
<code>u</code>	Numeric vector or matrix of reference points for <code>x</code> , or <code>NULL</code> (the default). Must have the same number of rows as <code>x</code> and the same number of columns as <code>x</code> (after coercion with <code>as.matrix()</code>). If <code>NULL</code> , the input must be one-dimensional and the uniform grid $\{1/n, \dots, 1\}$ is used.
<code>v</code>	Numeric vector or matrix of reference points for <code>y</code> , or <code>NULL</code> (the default). Must have the same number of rows as <code>y</code> and the same number of columns as <code>y</code> (after coercion with <code>as.matrix()</code>). If <code>NULL</code> , the input must be one-dimensional and the uniform grid $\{1/n, \dots, 1\}$ is used.
<code>visualise</code>	Logical; if <code>TRUE</code> , displays a scatter plot of the rank-transformed points with overlaid rectangles to illustrate the coverage calculation. The default is <code>FALSE</code> (no plot). If set to <code>TRUE</code> but either <code>x</code> or <code>y</code> has more than one column, a warning is issued and <code>visualise</code> is reset to <code>FALSE</code> .
<code>method</code>	Character string specifying the computation method. Options are "auto", "exact", or "approx". See Details.
<code>M</code>	Integer; Number of Monte Carlo integration sample points (used when <code>method</code> = "approx"). Optional.
<code>na.rm</code>	Logical; if <code>TRUE</code> , remove NA values before computation.

Details

When both `x` and `y` are one-dimensional, `u` and `v` may be omitted; in that case they default to the deterministic uniform grid $\{1/n, 2/n, \dots, 1\}$, which is the natural 1D reference set and makes the rank transformation deterministic.

The `method` argument controls how the computation is performed:

- "exact": Computes the exact value.
- "approx": Uses a Monte Carlo approximation with `M` sample points.
- "auto": Automatically selects a method based on the total number of columns in `x` and `y`: if more than 6, "approx" is used (with $M = \text{nrow}(x)^{1.5}$ if `M` is not provided); otherwise, "exact" is used.

When `na.rm = TRUE`, rows with NA in `x` or `y` are removed, and the corresponding rows of any user-supplied `u` and `v` are removed as well so that the reference grids stay aligned with the data. Default grids are constructed after NA removal.

Value

A list with four elements:

- `stat` – The numeric value of the coverage correlation coefficient.
- `pval` – The p-value, calculated using the exact variance under the null hypothesis of independence between `x` and `y`.
- `method` – A character string indicating the computation method used.
- `mc_se` – A numeric value. If method "approx" was used `mc_se` is the standard error of the Monte Carlo approximation, otherwise it is 0.

See Also[coverage_correlation](#)**Examples**

```

set.seed(1)
n <- 100
x <- runif(n)
y <- sin(3*x) + runif(n) * 0.01
# 1D: u and v default to the uniform grid {1/n, ..., 1}
coverage_correlation_grid(x, y, visualise = TRUE)

# supplying grids explicitly still works
u <- runif(n)
v <- runif(n)
coverage_correlation_grid(x, y, u, v)

```

coverage_correlation_K

Coverage-based Dependence Measure for K Random Vectors

Description

Computes the coverage correlation coefficient jointly across K input variables or vectors $X_1, \dots, X_K$, following exactly the same logic as [coverage_correlation](#) but generalised from a pair to an arbitrary number of inputs. The coefficient measures the joint dependence among the K inputs.

Usage

```

coverage_correlation_K(
  data,
  method = c("auto", "exact", "approx"),
  M = NULL,
  na.rm = TRUE
)

```

Arguments

data	A list of K ($K \geq 2$) numeric vectors or matrices, each with the same number of rows n . Each element is one of the inputs $X_1, \dots, X_K$. Vectors are coerced to one-column matrices.
method	Character string specifying the computation method. Options are "auto", "exact", or "approx". See Details.
M	Integer; Number of Monte Carlo integration sample points (used when method = "approx"). Optional.
na.rm	Logical; if TRUE, remove rows with NA values (across all inputs) before computation.

Details

The procedure is as follows:

1. Calculate the rank transformation of each input X_k .
2. Construct small cubes of volume n^{-1} centered at each rank-transformed point in $[0, 1]^d$, where $d = d_1 + \dots + d_K$.
3. Compute the total volume of the union of these cubes, intersected with $[0, 1]^d$.

The coverage correlation coefficient is then calculated based on this union volume.

For more details, please refer to the original paper: [the arXiv preprint](#).

The method argument controls how the computation is performed:

- "exact": Computes the exact value.
- "approx": Uses a Monte Carlo approximation with M sample points.
- "auto": Automatically selects a method based on the total number of columns across all inputs: if more than 6, "approx" is used (with $M = n^{1.5}$ if M is not provided); otherwise, "exact" is used.

Value

A list with four elements:

- stat – The numeric value of the coverage correlation coefficient.
- pval – The p-value, calculated using the exact variance under the null hypothesis of mutual independence among the inputs.
- method – A character string indicating the computation method used.
- mc_se – A numeric value. If method "approx" was used mc_se is the standard error of the Monte Carlo approximation, otherwise it is 0.

See Also

[coverage_correlation](#)

Examples

```
set.seed(1)
n <- 100
x1 <- runif(n)
x2 <- sin(3 * x1) + runif(n) * 0.01
x3 <- x1 * x2 + runif(n) * 0.1
coverage_correlation_K(list(x1, x2, x3))
```

coverage_correlation_K_grid

Coverage-based Dependence Measure for K Random Vectors with Optional Reference Grids

Description

Identical to `coverage_correlation_K`, except that the reference points used for the Monge–Kantorovich rank transformation may be supplied directly via `grid` rather than being generated internally from `runif`. This is useful for reproducibility, for using low-discrepancy (e.g. quasi-Monte Carlo) reference grids, or for comparing results across calls with a fixed reference set.

Usage

```
coverage_correlation_K_grid(
  data,
  grid = NULL,
  method = c("auto", "exact", "approx"),
  M = NULL,
  na.rm = TRUE
)
```

Arguments

<code>data</code>	A list of K ($K \geq 2$) numeric vectors or matrices, each with the same number of rows n . Each element is one of the inputs $X_1, \dots, X_K$. Vectors are coerced to one-column matrices.
<code>grid</code>	A list of K reference-point sets parallel to <code>data</code> , or <code>NULL</code> (the default). When supplied, the k -th element must have the same number of rows and the same number of columns as the k -th element of <code>data</code> (after coercion with <code>as.matrix()</code>). Individual elements may also be <code>NULL</code> to request the default grid for that input only. If <code>grid</code> (or an element of it) is <code>NULL</code> , the corresponding input must be one-dimensional and the uniform grid $\{1/n, \dots, 1\}$ is used.
<code>method</code>	Character string specifying the computation method. Options are "auto", "exact", or "approx". See Details.
<code>M</code>	Integer; Number of Monte Carlo integration sample points (used when <code>method = "approx"</code>). Optional.
<code>na.rm</code>	Logical; if <code>TRUE</code> , remove rows with NA values (across all inputs) before computation.

Details

When every input in `data` is one-dimensional, `grid` may be omitted; in that case each reference grid defaults to the deterministic uniform grid $\{1/n, 2/n, \dots, 1\}$, which is the natural 1D reference set and makes the rank transformation deterministic.

The `method` argument controls how the computation is performed:

- "exact": Computes the exact value.
- "approx": Uses a Monte Carlo approximation with M sample points.
- "auto": Automatically selects a method based on the total number of columns across all inputs: if more than 6, "approx" is used (with $M = n^{1.5}$ if M is not provided); otherwise, "exact" is used.

When `na.rm = TRUE`, rows with NA in any input are removed, and the corresponding rows of any supplied grids are removed as well so that the reference grids stay aligned with the data. Default grids are constructed after NA removal.

Value

A list with four elements:

- `stat` – The numeric value of the coverage correlation coefficient.
- `pval` – The p-value, calculated using the exact variance under the null hypothesis of mutual independence among the inputs, if $K > 2$ returns NA.
- `method` – A character string indicating the computation method used.
- `mc_se` – A numeric value. If method "approx" was used `mc_se` is the standard error of the Monte Carlo approximation, otherwise it is 0.

See Also

[coverage_correlation_K](#), [coverage_correlation_grid](#)

Examples

```
set.seed(1)
n <- 100
x1 <- runif(n)
x2 <- sin(3 * x1) + runif(n) * 0.01
x3 <- x1 * x2 + runif(n) * 0.1
# all 1D: grid defaults to the uniform grid {1/n, ..., 1} for each input
coverage_correlation_K_grid(list(x1, x2, x3))

# supplying grids explicitly still works
g <- list(runif(n), runif(n), runif(n))
coverage_correlation_K_grid(list(x1, x2, x3), grid = g)
```

covercorr

Coverage-based Dependence Measure (unified interface)

Description

A single front-end that dispatches to the appropriate coverage correlation routine based on its inputs. It bundles together [coverage_correlation](#), [coverage_correlation_grid](#), [coverage_correlation_K](#), and [coverage_correlation_K_grid](#).

Usage

```
covercorr(
  x,
  y = NULL,
  reference = c("random", "deterministic"),
  grid = NULL,
  visualise = FALSE,
  method = c("auto", "exact", "approx"),
  M = NULL,
  na.rm = TRUE
)
```

Arguments

<code>x</code>	A numeric vector, matrix, data frame, data table, or a list of numeric vectors/matrices. See Details for how each type is dispatched.
<code>y</code>	Optional numeric vector or matrix with the same number of rows as <code>x</code> . If supplied, the pairwise case is used and <code>x</code> is treated as a single (possibly multivariate) variable.
<code>reference</code>	Character string, either "random" (the default) or "deterministic", selecting the reference points. See Details.
<code>grid</code>	Optional list of reference-point sets, one per variable, each matching the shape of the variable it corresponds to (same number of rows and columns). In the pairwise case <code>grid</code> is a list of length two, <code>list(u, v)</code> , matching <code>x</code> and <code>y</code> respectively. In the K -variable case it is a list parallel to the variables. Supplying it implies <code>reference = "deterministic"</code> .
<code>visualise</code>	Logical; if TRUE <i>and</i> the inputs reduce to a pair of univariate variables, a scatter plot of the rank-transformed points with overlaid rectangles is displayed. Ignored (with a warning) otherwise. Default FALSE.
<code>method</code>	Character string specifying the computation method. Options are "auto", "exact", or "approx".
<code>M</code>	Integer; Number of Monte Carlo integration sample points (used when <code>method = "approx"</code>). Optional.
<code>na.rm</code>	Logical; if TRUE, remove rows with NA values before computation.

Details

Dispatch rules, based on the first argument `x`:

- If `y` is supplied, the pairwise case is used on (x, y) .
- If `x` is a list, each element is treated as one variable (possibly multivariate) and the K -input case is used.
- If `x` is a matrix, data frame, or data table (and `y` is not supplied), each *column* is treated as a univariate variable: a two-column input uses the pairwise case, and an input with more than two columns uses the K -input case with K equal to the number of columns.

The reference argument selects the reference points used for the Monge–Kantorovich rank transformation:

- "random": independent runif reference points are generated internally (routes to coverage_correlation / coverage_correlation_K).
- "deterministic": the deterministic uniform grid $\{1/n, \dots, 1\}$ is used for each (univariate) variable, unless the user supplies their own reference points via grid. Routes to coverage_correlation_grid / coverage_correlation_K_grid.

Supplying grid implies reference = "deterministic" automatically.

Value

A list with elements stat, pval, method, and mc_se; see [coverage_correlation](#) for details.

See Also

[coverage_correlation](#), [coverage_correlation_grid](#), [coverage_correlation_K](#), [coverage_correlation_K_grid](#)

Examples

```
set.seed(1)
n <- 100
x <- runif(n)
y <- sin(3 * x) + runif(n) * 0.01

# pairwise, random reference points
covercorr(x, y)

# pairwise, deterministic grid, with visualisation
covercorr(x, y, reference = "deterministic", visualise = TRUE)

# pairwise, user-supplied reference points (implies deterministic)
covercorr(x, y, grid = list(runif(n), runif(n)))

# K variables as a matrix; supply custom grids for each column
z <- x * y + runif(n) * 0.1
covercorr(cbind(x, y, z), grid = list(runif(n), runif(n), runif(n)))
```

covered_volume	<i>Total volume of union of rectangles</i>
----------------	--

Description

Total volume of union of rectangles

Usage

```
covered_volume(zmin, zmax)
```

Arguments

- zmin n x d matrix of bottomleft coordinates, one row per rectangle
- zmax n x d matrix of topright coordinates, one row per rectangle

Details

This is a wrapper of the C_covered_volume_partitioned function in C

Value

a numeric value of the volume of the union

covered_volume_mc	<i>Total volume of union of rectangles using Monte Carlo integration</i>
-------------------	--

Description

Total volume of union of rectangles using Monte Carlo integration

Usage

covered_volume_mc(zmin_s, zmax_s, M)

Arguments

- zmin_s n x d matrix of bottomleft coordinates, one row per rectangle
- zmax_s n x d matrix of topright coordinates, one row per rectangle
- M number of Monte Carlo integration sample points

Details

This is a wrapper of the C_covered_volume_mc function in C

Value

a list of the estimated volume of the union and its standard error

covered_volume_partitioned	<i>Total volume of union of rectangles using volume hashing</i>
----------------------------	---

Description

Total volume of union of rectangles using volume hashing

Usage

```
covered_volume_partitioned(zmin, zmax)
```

Arguments

zmin	n x d matrix of bottomleft coordinates, one row per rectangle
zmax	n x d matrix of topright coordinates, one row per rectangle

Details

This is a wrapper of the C_covered_volume_partitioned function in C

Value

a numeric value of the volume of the union

MK_rank	<i>Monge–Kantorovich ranks (uniform OT via squared distances)</i>
---------	---

Description

Computes the optimal matching that maps each observation in X to a reference point in U using uniform weights and squared Euclidean cost. Internally uses `transport::transport(method = "networkflow", p = 2)`. In 1D, this reduces to a rank-based matching `sort(U)[rank(X, ties.method = "random")]`.

Usage

```
MK_rank(X, U)
```

Arguments

X	Numeric vector of length n , or numeric matrix with n rows and d columns. If not a matrix, it is coerced with <code>as.matrix()</code> .
U	Numeric vector of length n , or numeric matrix with n rows and d columns. If not a matrix, it is coerced with <code>as.matrix()</code> . Must have the same number of rows as X .

Details

- Rows must match: `nrow(X) == nrow(U)` (otherwise an error is thrown).
- Columns must match: `ncol(X) == ncol(U)` (otherwise an error is thrown).
- Weights are uniform ($1/n$) and the cost matrix is the sum of squared coordinate differences across columns.
- In 1D, ties in X are broken at random via `ties.method = "random"`; use `set.seed()` for reproducibility.

Value

If `ncol(X) == 1`, a numeric vector of length n containing the entries of U reordered to match the ranks of X . Otherwise, a numeric $n \times d$ matrix whose i -th row is the matched row of U corresponding to the i -th row of X .

Dependencies

Requires the **transport** package.

Examples

```
# 1D example (set seed for reproducible tie-breaking)
set.seed(1)
x <- rnorm(10)
u <- seq(0, 1, length.out = 10)
MK_rank(x, u)

# 2D example
set.seed(42)
X <- matrix(rnorm(200), ncol = 2) # 100 x 2
U <- matrix(runif(200), ncol = 2) # 100 x 2
R <- MK_rank(X, U)
dim(R) # 100 2
```

plot_rectangles

Plot a collection of axis-aligned rectangles in the unit square

Description

Draws rectangles specified by their `xmin`, `xmax`, `ymin`, and `ymax`, optionally adding them to an existing plot. When `add = FALSE`, a fresh $[0, 1] \times [0, 1]$ plot with a grid and equal aspect ratio is created.

Usage

```
plot_rectangles(xmin, xmax, ymin, ymax, add = FALSE)
```

Arguments

xmin	Numeric vector of left x-coordinates.
xmax	Numeric vector of right x-coordinates (same length as xmin).
ymin	Numeric vector of bottom y-coordinates (same length as xmin).
ymax	Numeric vector of top y-coordinates (same length as xmin).
add	Logical; if TRUE, add to an existing plot. Default FALSE.

Value

Invisibly returns NULL. Use this function for its plotting output, not for a returned value.

split_rectangles	<i>Split rectangles by wrapping them around edges of $[0, 1]^d$</i>
------------------	--

Description

Split rectangles by wrapping them around edges of $[0, 1]^d$

Usage

```
split_rectangles(zmin, zmax)
```

Arguments

zmin	n x d matrix of bottom-left coordinates, one row per rectangle
zmax	n x d matrix of top-right coordinates, one row per rectangle

Details

This is a wrapper of the C_split_rectangles function implemented in C

Value

a list of zmin and zmax, describing the bottom-left and top-right coordinates of splitted rectangles

variance_formula	<i>Variance of the the excess vacancy</i>
------------------	---

Description

Exact formula for n times the variance of the excess vacancy. For independent X and Y , the variance of the coverage correlation coefficient is obtained by dividing the returned value by $n(1 - e^{-1})^2$. check [the arXiv preprint](#) for more details

Usage

```
variance_formula(n, d)
```

Arguments

n	sample size
d	dimension (X, Y)

Value

variance formula in paper

visualise_density	<i>Visualise the joint density of two rank vectors on the unit square</i>
-------------------	---

Description

Produces an image plot of the log-ratio of an estimated joint density to the uniform density over $[0, 1]^2$, given the (Monge–Kantorovich) rank transformations of two variables. Regions where the pair of variables is more concentrated than under independence appear in one colour and regions where it is sparser appear in another, making departures from independence easy to read off visually.

Usage

```
visualise_density(x_rank, y_rank)
```

Arguments

x_rank	Numeric vector of rank values in $[0, 1]$ for the first variable, e.g. the output of MK_rank .
y_rank	Numeric vector of rank values in $[0, 1]$ for the second variable, of the same length as x_rank.

Details

The density is estimated with a two-dimensional kernel density estimate ([kde2d](#)). To avoid edge effects near the boundary of the unit square, the points are first tiled toroidally (see the internal `wrap_points` helper) before the kernel density estimate is computed on $[0, 1]^2$. The estimate is normalised by its mean over the unit square, so that a value of zero on the log scale corresponds to the uniform density.

Value

Invisibly returns NULL. Use this function for its plotting output, not for a returned value.

See Also

[MK_rank](#), [coverage_correlation](#)

Examples

```
set.seed(1)
n <- 500
x <- runif(n)
y <- sin(3 * x) + runif(n) * 0.1
x_rank <- MK_rank(as.matrix(x), runif(n))
y_rank <- MK_rank(as.matrix(y), runif(n))
visualise_density(x_rank, y_rank)
```

Index

* datasets

CD8T, [2](#)

CD8T, [2](#)

coverage_correlation, [2](#), [4](#), [6](#), [7](#), [9](#), [11](#), [17](#)

coverage_correlation_grid, [4](#), [9](#), [11](#)

coverage_correlation_K, [6](#), [8](#), [9](#), [11](#)

coverage_correlation_K_grid, [8](#), [9](#), [11](#)

covercorr, [9](#)

covered_volume, [11](#)

covered_volume_mc, [12](#)

covered_volume_partitioned, [13](#)

kde2d, [17](#)

MK_rank, [13](#), [16](#), [17](#)

plot_rectangles, [14](#)

split_rectangles, [15](#)

variance_formula, [16](#)

visualise_density, [16](#)