

# Package ‘BlockwiseRankTest’

October 1, 2025

**Type** Package

**Title** Block-Wise Rank in Similarity Graph Edge-Count Two-Sample Test (BRISE)

**Version** 0.1.0

**Maintainer** Kejian Zhang <kejianzhang@u.nus.edu>

**Description** Implements the Block-wise Rank in Similarity Graph Edge-count test (BRISE), a rank-based two-sample test designed for block-wise missing data. The method constructs (pattern) pair-wise similarity graphs and derives quadratic test statistics with asymptotic chi-square distribution or permutation-based p-values. It provides both vectorized and congregated versions for flexible inference. The methodology is described in Zhang, Liang, Maile, and Zhou (2025) <[doi:10.48550/arXiv.2508.17411](https://doi.org/10.48550/arXiv.2508.17411)>.

**License** GPL (>= 2)

**Encoding** UTF-8

**Depends** R (>= 3.5.0)

**Imports** stats

**Suggests** testthat (>= 3.0.0), knitr, rmarkdown

**RoxygenNote** 7.3.3

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Kejian Zhang [aut, cre],  
Doudou Zhou [aut] (ORCID: <<https://orcid.org/0000-0002-0830-2287>>)

**Repository** CRAN

**Date/Publication** 2025-10-01 07:00:26 UTC

## Contents

BRISE . . . . .	2
<b>Index</b>	<b>4</b>

**Description**

BRISE implements the Two-Sample Test that handles block-wise missingness. It identifies missing-data patterns, constructs a (blockwise) dissimilarity matrix, induces ranks via a k-nearest neighbor style graph, and computes a quadratic statistic under two versions: the congregated form ('con') and vectorized form ('vec'). Permutation p-values are optionally available.

**Usage**

```
BRISE(
  X = NULL,
  Y = NULL,
  D = NULL,
  ptn_list = NULL,
  k = 10,
  perm = 0,
  skip = 1,
  ver = "con"
)
```

**Arguments**

X	Numeric matrix ( $m \times p$ ) of observations for X (Sample 1). Optional if D and ptn_list are provided.
Y	Numeric matrix ( $n \times p$ ) of observations for Y (Sample 2). Optional if D and ptn_list are provided.
D	Numeric square dissimilarity matrix ( $N \times N$ ), where $N = m + n$ . Required when X and Y are not given.
ptn_list	List of integer vectors. Each element contains indices ( $1 \dots N$ ) of observations that share the same missing-data pattern.
k	Positive integer. Neighborhood size offset for rank truncation in nearest-neighbor ranking. Default is 10.
perm	Integer. Number of permutations for computing permutation p-value. Default is 0 (no permutation).
skip	Integer (0 or 1). When set to 1 (default), skip rank-based dissimilarity for modality pairs with no shared observed variables; setting to 0 computes them (slower).
ver	Character. Version of the test statistic: "con" (congregated form, default) or "vec" (vectorized form).

## Details

If both  $X$  and  $Y$  are supplied, `Identify_mods` is used to detect missing patterns and reorganize variables by modality. The dissimilarity matrix  $D$  is then constructed via `Blockdist`. Patterns with too few observations in either sample (e.g. fewer than 2) or patterns that are very small relative to the largest pattern are filtered out for robustness. A symmetric rank matrix is built based on truncated nearest-neighbor ranks. Under `ver="con"` the contrast statistic (two degrees of freedom) is used; under `ver="vec"` a higher-dimensional vector statistic is used. Asymptotic p-values use chi-square approximations; if `perm > 0`, empirical permutation p-values are also computed.

## Value

A list with elements:

**test.statistic** Numeric. The computed test statistic.

**pval.approx** Numeric. Asymptotic p-value (chi-square based).

**Cov** Covariance matrix used in computing the test statistic.

**pval.perm** (Optional) Permutation p-value if `perm > 0`.

## References

Zhang, K., Liang, M., Maile, R. & Zhou, D. (2025). *Two-Sample Testing with Block-wise Missingness in Multi-source Data*. *arXiv preprint arXiv:2508.17411*.

## See Also

[BRISE\\_Rank](#), [Cov\\_mu.c](#), [Cov\\_mu.v](#)

## Examples

```
set.seed(1)
X <- matrix(rnorm(50*200, mean = 0), nrow=50)
Y <- matrix(rnorm(50*200, mean = 0.3), nrow=50)
X[1:20, 1:100] <- 0
X[30:50, 101:200] <- 0
Y[1:10, 1:100] <- 0
Y[30:40, 101:200] <- 0
out <- BRISE(X = X, Y = Y, k = 5, perm = 1000, ver = "con")
print(out$test.statistic)
print(out$pval.approx)
```

# Index

BRISE, [2](#)  
BRISE\_Rank, [3](#)  
  
Cov\_mu.c, [3](#)  
Cov\_mu.v, [3](#)