

Package: SimIndep (via r-universe)

May 13, 2026

Type Package

Title WISE: a Weighted Similarity Aggregation Test for Serial Independence

Version 0.1.2

Description A fast implementation of the weighted information similarity aggregation (WISE) test for detecting serial dependence, particularly suited for high-dimensional and non-Euclidean time series. Includes functions for constructing similarity matrices and conducting hypothesis testing. Users can use different similarity measures and define their own weighting schemes. For more details see Q Zhu, M Liu, Y Han, D Zhou (2025) <[doi:10.48550/arXiv.2509.05678](https://doi.org/10.48550/arXiv.2509.05678)>.

Imports FNN, stats

Suggests MASS

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation no

Author Qihua Zhu [aut, cre], Mingshuo Liu [aut], Yuefeng Han [aut], Doudou Zhou [aut]

Maintainer Qihua Zhu <zhuqihua@u.nus.edu>

Repository <https://cran.r-universe.dev>

Date/Publication 2025-10-15 19:30:10 UTC

RemoteUrl <https://github.com/cran/SimIndep>

RemoteRef HEAD

RemoteSha e0ee9fa0099c5c8e252ece3e336a9034106ce319

Contents

wise_sim	2
wise_test	2

Index**5**

wise_sim	<i>Calculate an n by n similarity matrix</i>
----------	--

Description

Returns an n by n similarity matrix.

Usage

```
wise_sim(data, measure = "distance", metric = "manhattan", k = NULL)
```

Arguments

data	an n by p data matrix, with n being the sample size and p being the dimension.
measure	the similarity measure: "distance" for distance-based measure; "graph" for k-nearest neighbor graph-based measure. The default is "distance".
metric	character string specifying the distance metric or graph weight. "manhattan" for Manhattan distance (default), "euclidean" for Euclidean distance.
k	the Number of nearest neighbors used in k-nearest neighbor graph. k = floor(sqrt(n)) if not specified.

Value

an n by n similarity matrix.

Examples

```
X <- matrix(rnorm(100), nrow = 10)
wise_sim(X, measure = "distance", metric = "manhattan")
```

wise_test	<i>Conducts the serial independence test (WISE) based on a similarity matrix</i>
-----------	--

Description

Returns the p-value of WISE, the squared test statistic, and related quantities (the chi-square critical value, permutation mean, permutation variance).

Usage

```
wise_test(sim, dependence = "proximity", alpha = 0.05, weight = NULL, h = 4)
```

Arguments

sim	an n by n similarity matrix, typically generated from wise_sim().
dependence	design for the weight matrix W: if "proximity", $W_{ij} = (1/(i - j ^2 + 1)) - 1$; if "periodicity", then $W_{ij} = \cos(i - j \pi/h) - 1$; If "customized", users should input their self-defined weight matrix through the parameter "weight". The default is "proximity"
alpha	the nominal significance level (default is 0.05).
weight	an n by n weight matrix with zero diagonal (only used if dependence = "customized").
h	the estimated periodicity (default is 4). The parameter is used only if dependence = "periodicity".

Value

A list containing:

p_value	The p-value of the test.
test_statistic_sq	The value of the squared test statistic.
critical_value	The chi-square critical value at the given significance level.
t	The unstandardized test statistic.
permutation_mean	The mean of t under the permutation null.
permutation_variance	The variance of t under the permutation null.

Examples

```
library(MASS)
n <- 100
p <- 50

# Example 1: iid data
set.seed(123)
data_iid <- mvrnorm(n = n, mu = rep(0, p) , Sigma = diag(p))
wise_test(
  wise_sim(data_iid, measure = "distance", metric = "manhattan"),
  dependence = "proximity",
  alpha = 0.05
)

# Example 2: AR(1)
set.seed(123)
data_ar <- matrix(0, nrow = n, ncol = p)
error <- mvrnorm(n = n, mu = rep(0,p), Sigma = diag(p))
data_ar[1,] <- error[1,]
phi <- 0.1 * diag(p)
for (t in 2:n) {
```

```
    data_ar[t, ] <- phi %% data_ar[t - 1, ] + error[t,]
  }
  wise_test(
    wise_sim(data_ar, measure = "distance", metric = "manhattan"),
    dependence = "proximity",
    alpha = 0.05
  )

# Example 3: NMA(2)
set.seed(123)
data_nma <- matrix(0, nrow = n, ncol = p)
error <- mvrnorm(n = n, mu = rep(0,p), Sigma = diag(p))
data_nma[1:2, 1:p] <- error[1:2,1:p]
for (i in 3:n) {
  data_nma[i, ] <- error[i,]*error[i-1,]*error[i-2,]
}
wise_test(
  wise_sim(data_nma, measure = "distance", metric = "manhattan"),
  dependence = "proximity",
  alpha = 0.05
)
```

Index

wise_sim, 2
wise_test, 2