

Package ‘spots’

May 3, 2022

Title Spatial Component Analysis

Version 0.1.0

Description The spots package is designed for spatial omics (10x Visium, etc.) data analysis. It performs various statistical analyses and tests, including spatial component analysis (SCA), both global and local spatial statistics, such as univariate and bivariate Moran's I, Getis-Ord G_i^* statistics, etc. See Integrated protein and transcriptome high-throughput spatial profiling (2022) <[doi:10.1101/2022.03.15.484516](https://doi.org/10.1101/2022.03.15.484516)> for more details.

Maintainer X. Steve Niu <xin2001@med.cornell.edu>

BugReports <https://github.com/steveXniu/spots/issues>

Depends R (>= 4.0.0)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Collate 'data.R' 'stats.R' 'hnn.R' 'local.R' 'sca.R' 'moransi.R'
'zzz.R'

Imports Matrix, RSpecra, tools, utils, methods, stats

Suggests ape, knitr, rmarkdown, testthat (>= 3.0.0), covr

Config/testthat/edition 3

NeedsCompilation no

Author X. Steve Niu [aut, cre] (<<https://orcid.org/0000-0001-7689-0941>>)

Repository CRAN

Date/Publication 2022-05-03 10:10:02 UTC

R topics documented:

spots-package	2
available.data	3
BivariateMoransI	3
BivariateMoransIStats	4

FitMoransI	5
HnnImpute	6
HnnNeighbor	7
HnnWeight	8
IrregularData	8
IrregularMoransI	9
LeeL	10
LoadData	11
LocalGi	12
LocalMoransI	13
LocalMoransIStats	14
OLSMoransI	15
PermutationCorr	16
PermutationGi	17
PermutationLocalI	18
PermutationMoransI	20
SCA	21
SpatialXCorr	22
UnivariateMoransI	23
VisiumHnn	24
WartenbergM	25
ZPvalue	26
Index	27

 spots-package

spots: Spatial Component Analysis

Description

The spots package is designed for spatial omics (10x Visium, etc.) data analysis. It performs various statistical analyses and tests, including spatial component analysis (SCA), both global and local spatial statistics, such as univariate and bivariate Moran's I, Getis-Ord Gi* statistics, etc. See Integrated protein and transcriptome high-throughput spatial profiling (2022) <doi:10.1101/2022.03.15.484516> for more details.

Author(s)

Maintainer: X. Steve Niu <xin2001@med.cornell.edu> ([ORCID](#))

See Also

Useful links:

- Report bugs at <https://github.com/stevexniu/spots/issues>

available.data	<i>Available data</i>
----------------	-----------------------

Description

Available data can be loaded using [LoadData](#) function.

Usage

```
available.data
```

Format

A data frame with 2 rows (datasets) and 5 columns (attributes):

filename Dataset

md5 MD5 checksum

url URL link

date Date updated

size Data size

BivariateMoransI	<i>Bivariate Moran's I</i>
------------------	----------------------------

Description

Calculate bivariate (multivariate) Moran's I using Wartenberg's method.

Usage

```
BivariateMoransI(  
  X,  
  W,  
  alternative = c("two.sided", "less", "greater"),  
  p.adjust.method = "BH"  
)
```

Arguments

X A matrix with observations as rows and features as columns.

W A weight matrix across all observations, i.e inverse of a pairwise distance matrix.

alternative Alternative hypothesis used, default is two.sided.

p.adjust.method Method used for multiple comparisons correction, default is BH. See [p.adjust](#).

Value

A list containing the following:

- `Morans.I`, the Moran's I.
- `Z.I`, the Z score of Moran's I.
- `Expected.I`, the expectation of Moran's I under the null hypothesis.
- `SD.I`, the standard deviation of Moran's I under the null hypothesis.
- `alternative`, alternative hypothesis used.
- `p.adjust.method`, method used for multiple comparisons correction.

References

Wartenberg, D. Multivariate spatial correlation: A method for exploratory geographical analysis. *Geogr. Anal.* 17, 263–283 (1985)

Czaplewski, R. L. Expected Value and Variance of Moran's Bivariate Spatial Autocorrelation Statistic for a Permutation Test. (U.S. Department of Agriculture, Forest Service, Rocky Mountain Forest and Range Experiment Station, 1993).

Examples

```
{
data.use <- quakes[1:100,]
W <- 1/as.matrix(dist(data.use[,1:2]))
diag(W) <- 0
res <- BivariateMoransI(data.use[,3:4], W)
}
```

BivariateMoransIStats *Bivariate Moran's I statistics*

Description

Calculate theoretical expectation and standard deviation of bivariate Moran's I under the null hypothesis.

Usage

```
BivariateMoransIStats(X, W)
```

Arguments

`X` A matrix with observations as rows and features as columns.

`W` A weight matrix across all observations, i.e inverse of a pairwise distance matrix.

Value

A list containing the following:

- E.I, the expectation of Moran's I under the null hypothesis.
- SD.I, the standard deviation of Moran's I under the null hypothesis.

References

Czaplewski, R. L. Expected Value and Variance of Moran's Bivariate Spatial Autocorrelation Statistic for a Permutation Test. (U.S. Department of Agriculture, Forest Service, Rocky Mountain Forest and Range Experiment Station, 1993).

Examples

```
{
data.use <- quakes[1:100,]
W <- 1/as.matrix(dist(data.use[,1:2]))
diag(W) <- 0
res <- BivariateMoransIStats(data.use[,3:4], W)
}
```

FitMoransI

Fitted values of Moran's I

Description

Calculate fitted values of Moran's I as linear regression.

Usage

```
FitMoransI(X, Y, I, all = FALSE)
```

Arguments

X	A matrix with observations as rows and features as columns.
Y	A matrix of spatial lags.
I	A matrix of Moran's I, output by OLSMoransI..
all	Whether to use multivariate Moran's I, default is FALSE.

Value

A list containing the following:

- residuals, residuals of fitted Moran's I.
- fitted.values, fitted values of Moran's I.
- all.fitted.values, a three-dimensional tensor of all fitted multivariate Moran's I, if calculated.
- all.residuals, a three-dimensional tensor of all residuals using fitted multivariate Moran's I, if calculated.

References

Anselin, L. Local indicators of spatial association-LISA. Geogr. Anal. 27, 93–115 (2010)

Examples

```
{
data.use <- quakes[1:100,]
W <- 1/as.matrix(dist(data.use[,1:2]))
diag(W) <- 0
res <- OLSMoransI(data.use[,3:4], W)
res.fitted <- FitMoransI(res$X, res$Y, res$Morans.I)
}
```

HnnImpute

Hexagonal nearest neighbor based imputation

Description

Data imputation and smoothing using hexagonal nearest neighbor.

Usage

```
HnnImpute(data, dist.hnn, dist.k = NULL, mu = 0, sigma = 1)
```

Arguments

data	A data matrix with features as rows and observations as columns.
dist.hnn	A hexagonal nearest neighbor distance matrix.
dist.k	The maximum distance used to calculate the weight. Default is NULL and all neighbor weights are calculated.
mu	The mean of Gaussian filter, default is 0.
sigma	The standard deviation of Gaussian filter, default is 1.

Value

Imputed data.

See Also

[HnnWeight](#)

Examples

```
{
data.use <- quakes[1:100,]
dist.use <- as.matrix(dist(data.use[,1:2]))
# transpose the data to have features in rows and observations in columns
res <- HnnImpute(t(data.use[,3:4]), dist.use)
}
```

HnnNeighbor*Hexagonal nearest neighbor*

Description

Calculate hexagonal nearest neighbors.

Usage

```
HnnNeighbor(dist.hnn, k, include.self = TRUE)
```

Arguments

`dist.hnn` A hexagonal nearest neighbor distance matrix.
`k` Number of neighbors.
`include.self` Whether to include self as 1st neighbor, default is TRUE.

Value

A list containing the following:

- `knn.idx`, an $n \times k$ matrix for the nearest neighbor indice.
- `knn.dist`, an $n \times k$ matrix for the nearest neighbor hexagonal distances.
- `dist.mat`, a connectivity-based distance matrix.

References

Middleton, L. & Sivaswamy, J. Edge detection in a hexagonal-image processing framework. *Image Vis. Comput.* 19, 1071–1081 (2001)

Examples

```
{
data.use <- quakes[1:100,]
dist.use <- as.matrix(dist(data.use[,1:2]))
res <- HnnNeighbor(dist.use, k = 10)
}
```

HnnWeight	<i>Hexagonal nearest neighbor weight</i>
-----------	--

Description

Calculate hexagonal nearest neighbor weights using Gaussian filter.

Usage

```
HnnWeight(dist.hnn, dist.k = NULL, mu = 0, sigma = 1)
```

Arguments

dist.hnn	A hexagonal nearest neighbor distance matrix.
dist.k	The maximum distance used to calculate the weight. Default is NULL and all neighbor weights are calculated.
mu	The mean of Gaussian filter, default is 0.
sigma	The standard deviation of Gaussian filter, default is 1.

Value

A weight matrix.

Examples

```
{
data.use <- quakes[1:100,]
dist.use <- as.matrix(dist(data.use[,1:2]))
res <- HnnWeight(dist.use, mu = 0, sigma = 0.5)
}
```

IrregularData	<i>Process irregular data shape (between two groups) for Moran's I</i>
---------------	--

Description

Process irregular data shape for Moran's I between two groups.

Usage

```
IrregularData(X, W, group1, group2)
```


Arguments

X	A matrix with observations as rows and features as columns.
W	A weight matrix across all observations, i.e inverse of a pairwise distance matrix.
group1	The indices or names for the first group of observations.
group2	The indices or names for the second group of observations.

Value

A list containing the following:

- X, new data matrix of group1 and group2.
- W, new weight matrix of group1 and group2.

Examples

```
{
data.use <- quakes[1:100,]
W <- 1/as.matrix(dist(data.use[,1:2]))
diag(W) <- 0
res <- IrregularData(data.use[,3:4], W, 1:10, 21:30)
}
```

IrregularMoransI	<i>Moran's I between two groups (irregular data shape)</i>
------------------	--

Description

Calculate Moran's I between two groups with irregular data shape.

Usage

```
IrregularMoransI(
  X,
  W,
  group1,
  group2,
  OLS = FALSE,
  normalize = TRUE,
  alternative = c("two.sided", "less", "greater"),
  p.adjust.method = "BH"
)
```

Arguments

X	A matrix with observations as rows and features as columns.
W	A weight matrix across all observations, i.e inverse of a pairwise distance matrix.
group1	The indices or names for the first group of observations.
group2	The indices or names for the second group of observations.
OLS	Whether to use <code>OLSMoransI</code> , default is FALSE and <code>BivariateMoransI</code> is used.
normalize	Whether to normalize the weight matrix such that each row adds up to one. Default is TRUE.
alternative	Alternative hypothesis used, default is <code>two.sided</code> .
p.adjust.method	Method used for multiple comparisons correction, default is <code>BH</code> . See <code>p.adjust</code> .

Value

A list containing the output from `BivariateMoransI` or `OLSMoransI`.

Examples

```
{
data.use <- quakes[1:100,]
W <- 1/as.matrix(dist(data.use[,1:2]))
diag(W) <- 0
res <- IrregularMoransI(data.use[,3:4], W, 1:10, 21:30)
}
```

 LeeL

Bivariate Moran's I using Lee's L measurement

Description

Calculate bivariate Moran's I using Lee's L measurement.

Usage

```
LeeL(X, W)
```

Arguments

X	A matrix with observations as rows and features as columns.
W	A weight matrix across all observations, i.e inverse of a pairwise distance matrix.

Value

A spatial cross-correlation matrix.

References

Lee, S.-I. Developing a bivariate spatial association measure: An integration of Pearson's r and Moran's I . *J. Geogr. Syst.* 3, 369–385 (2001)

Examples

```
{
  data.use <- quakes[1:100,]
  W <- 1/as.matrix(dist(data.use[,1:2]))
  diag(W) <- 0
  L <- Leel(data.use[,3:4], W)
}
```

 LoadData

Load selected data

Description

If not previously run, it will download the selected data.

Usage

```
LoadData(file.path, data.use = c("Visium.HNN", "Visium.Brain"))
```

Arguments

file.path	Path to the directory contains or to download the data object, i.e. "~/Downloads".
data.use	Selected data to load. See available.data . <ul style="list-style-type: none"> Visium.HNN, hexagonal nearest neighbor distance matrix of 10x Visium 4992 whitelist spatial barcodes. Visium.Brain, sample 10x Visium Sagittal-Posterior Mouse Brain data with 3353 spatial barcodes and 2000 highly variable genes, load as a Seurat object.

Value

Selected data:

Examples

```
## Not run:
Visium.hnn.dist <- LoadData("~/Downloads", "Visium.HNN")

## End(Not run)
```

LocalGi*Local Getis-Ord Gi or Gi* statistics for spatial hot spot analysis*

Description

Calculate local Getis-Ord Gi or Gi* statistics.

Usage

```
LocalGi(  
  X,  
  W,  
  gstar = TRUE,  
  alternative = c("two.sided", "less", "greater"),  
  p.adjust.method = "BH"  
)
```

Arguments

X	A matrix with observations as rows and features as columns.
W	A weight matrix across all observations, i.e inverse of a pairwise distance matrix.
gstar	Whether to calculate the Gi* statistics, default is TRUE.
alternative	Alternative hypothesis used, default is two.sided.
p.adjust.method	Method used for multiple comparisons correction, default is BH. See p.adjust .

Value

A list containing the following:

- Gi, Gi or Gi* statistics.
- p.val, permutation based p-value.
- p.adj, adjusted p-values.
- gstar, permutation used if returned.
- alternative, alternative hypothesis used.
- p.adjust.method, method used for multiple comparisons correction.

References

Getis, A. & Ord, J. K. The analysis of spatial association by use of distance statistics. *Geogr. Anal.* 24, 189–206 (1992)

Examples

```
{
data.use <- quakes[1:100,]
W <- 1/as.matrix(dist(data.use[,1:2]))
diag(W) <- 0
res <- LocalGi(data.use[,3:4], W)
}
```

LocalMoransI

*Local Moran's I***Description**

Calculate Local Moran's I.

Usage

```
LocalMoransI(
  X,
  W,
  normalize = TRUE,
  alternative = c("two.sided", "less", "greater"),
  p.adjust.method = "BH",
  scale = TRUE
)
```

Arguments

X	A matrix with observations as rows and features as columns.
W	A weight matrix across all observations, i.e inverse of a pairwise distance matrix.
normalize	Whether to normalize the weight matrix such that each row adds up to one. Default is TRUE.
alternative	Alternative hypothesis used, default is two.sided.
p.adjust.method	Method used for multiple comparisons correction, default is BH. See p.adjust .
scale	Whether to scale the data. Default is TRUE.

Value

A list containing the following:

- Local.Morans.I, the local Moran's I.
- Z.I, the Z score of local Moran's I.
- X, data matrix used for calculating local Moran's I.
- Y, a matrix of spatial lags.

- Expected.I, the expectation of Moran's I under the null hypothesis.
- SD.I, the standard deviation of Moran's I under the null hypothesis.
- p.val, p-values.
- p.adj, adjusted p-values.
- normalize, whether to normalize the weight matrix.
- scale, whether to scale the data.
- scale.factor, number of observations.
- alternative, alternative hypothesis used.
- p.adjust.method, method used for multiple comparisons correction.

References

Anselin, L. Local indicators of spatial association-LISA. *Geogr. Anal.* 27, 93–115 (1995) Sokal, R. R., Oden, N. L. & Thomson, B. A. Local spatial autocorrelation in a biological model. *Geogr. Anal.* 30, 331–354 (1998)

Examples

```
{
data.use <- quakes[1:100,]
W <- 1/as.matrix(dist(data.use[,1:2]))
diag(W) <- 0
res <- LocalMoransI(data.use[,3:4], W)
}
```

LocalMoransIStats *Local Moran's I statistics*

Description

Calculate theoretical expectation and standard deviation of local Moran's I under the null hypothesis.

Usage

```
LocalMoransIStats(X, W, scale = FALSE)
```

Arguments

X	A matrix with observations as rows and features as columns.
W	A weight matrix across all observations, i.e inverse of a pairwise distance matrix.
scale	Whether to scale the data. Default is FALSE.

Value

A list containing the following:

- E.I, the expectation of local Moran's I under the null hypothesis.
- SD.I, the standard deviation of local Moran's I under the null hypothesis.

References

Anselin, L. Local indicators of spatial association-LISA. *Geogr. Anal.* 27, 93–115 (1995)

Sokal, R. R., Oden, N. L. & Thomson, B. A. Local spatial autocorrelation in a biological model. *Geogr. Anal.* 30, 331–354 (1998)

Examples

```
{
data.use <- quakes[1:100,]
W <- 1/as.matrix(dist(data.use[,1:2]))
diag(W) <- 0
res <- LocalMoransIStats(data.use[,3:4], W)
}
```

OLSMoransI

Moran's I using ordinary least squares (OLS)

Description

Calculate Moran's I as linear regression using ordinary least squares (OLS).

Usage

```
OLSMoransI(
  X,
  W,
  normalize = TRUE,
  alternative = c("two.sided", "less", "greater"),
  p.adjust.method = "BH"
)
```

Arguments

X	A matrix with observations as rows and features as columns.
W	A weight matrix across all observations, i.e inverse of a pairwise distance matrix.
normalize	Whether to normalize the weight matrix such that each row adds up to one. Default is TRUE.
alternative	Alternative hypothesis used, default is two.sided.
p.adjust.method	Method used for multiple comparisons correction, default is BH. See p.adjust .

Value

A list containing the following:

- Morans.I, the Moran's I.
- Z.I, the Z score of Moran's I.
- X, data matrix used for calculating Moran's I.
- Y, a matrix of spatial lags.
- Expected.I, the expectation of Moran's I under the null hypothesis.
- SD.I, the standard deviation of Moran's I under the null hypothesis.
- p.val, p-values.
- p.adj, adjusted p-values.
- alternative, alternative hypothesis used.
- p.adjust.method, method used for multiple comparisons correction.

References

Anselin, L. Local indicators of spatial association-LISA. *Geogr. Anal.* 27, 93–115 (2010)

Examples

```
{
data.use <- quakes[1:100,]
W <- 1/as.matrix(dist(data.use[,1:2]))
diag(W) <- 0
res <- OLSMoransI(data.use[,3:4], W)
}
```

PermutationCorr

Permutation based p-value for Pearson's correlation coefficient

Description

Calculate permutation based p-value for Pearson's correlation coefficient.

Usage

```
PermutationCorr(x, y, R = 999, seed = 1, return.permutation = FALSE)
```

Arguments

x	A numerical vector with the first variable.
y	A numerical vector with the second variable.
R	The number of permutations to be conducted, set to 999 by default.
seed	Random seed used. Default is 1.
return.permutation	Return permutations. Default is FALSE.

Details

It's adapted from `permcov` function in `Rfast` with corrections in calculating the number of permutations and the p-values.

Value

A list containing the following:

- correlation, Pearson's correlation coefficient.
- p.val, permutation based p-value.
- return.permutation, permutation used if returned.

Examples

```
{
x <- iris[, 1]
y <- iris[, 2]
res <- PermutationCorr(x, y, R = 9999)
}
```

PermutationGi	<i>Permutation based Getis-Ord Gi or Gi* statistics for spatial hot spot analysis</i>
---------------	---

Description

Calculate permutation based p-value for Getis-Ord Gi or Gi* statistics.

Usage

```
PermutationGi(
  x,
  W,
  gstar = TRUE,
  n = 999,
  seed = 1,
  alternative = c("two.sided", "less", "greater"),
  p.adjust.method = "BH",
  return.permutation = FALSE,
  condition = 0
)
```

Arguments

<code>x</code>	A numerical vector.
<code>W</code>	A weight matrix across all observations, i.e inverse of a pairwise distance matrix.
<code>gstar</code>	Whether to calculate the G_i^* statistics, default is TRUE.
<code>n</code>	The number of permutations to be conducted, set to 999 by default.
<code>seed</code>	Random seed used. Default is 1.
<code>alternative</code>	Alternative hypothesis used, default is <code>two.sided</code> .
<code>p.adjust.method</code>	Method used for multiple comparisons correction, default is BH. See p.adjust .
<code>return.permutation</code>	Return permutations. Default is FALSE.
<code>condition</code>	Value under null hypothesis to compare with, default is 0.

Value

A list containing the following:

- G_i , G_i or G_i^* statistics.
- `p.val`, permutation based p-value.
- `p.adj`, adjusted p-values.
- `return.permutation`, permutation used if returned.
- `params`, parameters used to calculate local Moran's I.

Examples

```
{
data.use <- quakes[1:100,]
W <- 1/as.matrix(dist(data.use[,1:2]))
diag(W) <- 0
res <- PermutationGi(data.use[,3], W)
}
```

PermutationLocalI

Permutation based local Moran's I

Description

Calculate permutation based p-value for local Moran's I.

Usage

```
PermutationLocalI(
  x,
  W,
  n = 999,
  seed = 1,
  alternative = c("two.sided", "less", "greater"),
  p.adjust.method = "BH",
  normalize = TRUE,
  scale = TRUE,
  return.permutation = FALSE,
  condition = 0
)
```

Arguments

x	A numerical vector.
W	A weight matrix across all observations, i.e inverse of a pairwise distance matrix.
n	The number of permutations to be conducted, set to 999 by default.
seed	Random seed used. Default is 1.
alternative	Alternative hypothesis used, default is two.sided.
p.adjust.method	Method used for multiple comparisons correction, default is BH. See p.adjust .
normalize	Whether to normalize the weight matrix such that each row adds up to one. Default is TRUE.
scale	Whether to scale the data. Default is TRUE.
return.permutation	Return permutations. Default is FALSE.
condition	Value under null hypothesis to compare with, default is 0.

Value

A list containing the following:

- Local.Morans.I, local Moran's I.
- p.val, permutation based p-value.
- p.adj, adjusted p-values.
- scale.factor, number of observations.
- return.permutation, permutation used if returned.
- params, parameters used to calculate local Moran's I.

Examples

```
{
data.use <- quakes[1:100,]
W <- 1/as.matrix(dist(data.use[,1:2]))
diag(W) <- 0
res <- PermutationLocalI(data.use[,3], W)
}
```

PermutationMoransI *Permutation based Moran's I*

Description

Calculate permutation based p-value for Moran's I.

Usage

```
PermutationMoransI(
  x,
  W,
  n = 999,
  seed = 1,
  normalize = TRUE,
  return.permutation = FALSE
)
```

Arguments

x	A numerical vector.
W	A weight matrix across all observations, i.e inverse of a pairwise distance matrix.
n	The number of permutations to be conducted, set to 999 by default.
seed	Random seed used. Default is 1.
normalize	Whether to normalize the weight matrix such that each row adds up to one. Default is TRUE.
return.permutation	Return permutations. Default is FALSE.

Value

A list containing the following:

- Morans.I, the Moran's I.
- p.val, permutation based p-value.
- return.permutation, permutation used if returned.

Examples

```
{
data.use <- quakes[1:100,]
W <- 1/as.matrix(dist(data.use[,1:2]))
diag(W) <- 0
res <- PermutationMoransI(data.use[,3], W)
}
```

SCA

*Spatial Component Analysis***Description**

Performs spatial component analysis (SCA) on the given data and weight matrices.

Usage

```
SCA(X, W, n.eigen = 20, method = c("L", "M"), scaled.data = NULL, ...)
```

Arguments

X	A matrix with observations as rows and features as columns.
W	A weight matrix across all observations, i.e inverse of a pairwise distance matrix.
n.eigen	Number of spatial components (eigenvectors) to compute. Default is 20.
method	Method used to calculate spatial cross-correlation. See SpatialXCorr . <ul style="list-style-type: none"> • M, using the Wartenburg's M (Default). • L, using the Lee's L.
scaled.data	Centered and scaled data used for SVD. Default is NULL.
...	Additional arguments passed for eigenvalue decomposition. See eigs_sym .

Value

A list of Spatial Component Analysis results.

- X, raw or scaled input data.
- rotation, computed eigenvectors.
- eigenvalues, computed eigenvalues.
- xcor, spatial cross-correlation matrix calculated using [SpatialXCorr](#).

References

Wartenberg, D. Multivariate spatial correlation: A method for exploratory geographical analysis. *Geogr. Anal.* 17, 263–283 (1985)

Lee, S.-I. Developing a bivariate spatial association measure: An integration of Pearson's r and Moran's I. *J. Geogr. Syst.* 3, 369–385 (2001)

Examples

```
{
data.use <- quakes[1:100,]
W <- 1/as.matrix(dist(data.use[,1:2]))
diag(W) <- 0
sca.res <- SCA(data.use[,3:5], W, n.eigen = 2)
}
```

SpatialXCorr*Spatial cross-correlation*

Description

Calculate spatial cross-correlation matrix.

Usage

```
SpatialXCorr(X, W, method = c("M", "L"))
```

Arguments

X	A matrix with observations as rows and features as columns.
W	A weight matrix across all observations, i.e inverse of a pairwise distance matrix.
method	Method used to calculate spatial cross-correlation. <ul style="list-style-type: none">• M, using the Wartenburg's M (Default).• L, using the Lee's L.

Value

A spatial cross-correlation matrix.

References

Wartenberg, D. Multivariate spatial correlation: A method for exploratory geographical analysis. *Geogr. Anal.* 17, 263–283 (1985)

Lee, S.-I. Developing a bivariate spatial association measure: An integration of Pearson's r and Moran's I. *J. Geogr. Syst.* 3, 369–385 (2001)

See Also

[WartenbergM LeelL.](#)

Examples

```
{
data.use <- quakes[1:100,]
W <- 1/as.matrix(dist(data.use[,1:2]))
diag(W) <- 0
M <- SpatialXCorr(data.use[,3:4], W, method = "M")
L <- SpatialXCorr(data.use[,3:4], W, method = "L")
}
```

UnivariateMoransI *Univariate Moran's I*

Description

Calculate univariate (canonical) Moran's I.

Usage

```
UnivariateMoransI(
  X,
  W,
  normalize = TRUE,
  alternative = c("two.sided", "less", "greater"),
  p.adjust.method = "BH"
)
```

Arguments

X	A matrix with observations as rows and features as columns.
W	A weight matrix across all observations, i.e inverse of a pairwise distance matrix.
normalize	Whether to normalize the weight matrix such that each row adds up to one. Default is TRUE.
alternative	Alternative hypothesis used, default is two.sided.
p.adjust.method	Method used for multiple comparisons correction, default is BH. See p.adjust .

Value

A list containing the following:

- Morans.I, the Moran's I.
- Z.I, the Z score of Moran's I.
- X, data matrix used for calculating Moran's I.
- Y, a matrix of spatial lags.
- Expected.I, the expectation of Moran's I under the null hypothesis.

- SD.I, the standard deviation of Moran's I under the null hypothesis.
- p.val, p-values.
- p.adj, adjusted p-values.
- normalize, whether to normalize the weight matrix.
- alternative, alternative hypothesis used.
- p.adjust.method, method used for multiple comparisons correction.

References

Moran, P. A. P. Notes on continuous stochastic phenomena. *Biometrika* 37, 17–23 (1950)

Examples

```
{
data.use <- quakes[1:100,]
W <- 1/as.matrix(dist(data.use[,1:2]))
diag(W) <- 0
res <- UnivariateMoransI(data.use[,3:4], W)
}
```

VisiumHnn

10x Genomics Visium hexagonal nearest neighbor

Description

Get hexagonal nearest neighbor distance between 10x Visium spatial barcodes.

Usage

```
VisiumHnn(path, barcodes)
```

Arguments

path	Path to the directory contains or to download the HNN distance for Visium barcodes, i.e. "~/Downloads".
barcodes	10x Visium whitelisted spatial barcodes.

Value

A hexagonal distance matrix for selected barcodes.

See Also

[available.data LoadData](#).

Examples

```
## Not run:
barcodes <- c("AAACAACGAATAGTTC-1", "AACTTAATTGCACGC-1")
VisiumHnn("~/Downloads/", barcodes)

## End(Not run)
```

WartenbergM

Bivariate Moran's I using Wartenberg's M matrix

Description

Calculate bivariate Moran's I using Wartenberg's method.

Usage

```
WartenbergM(X, W)
```

Arguments

X A matrix with observations as rows and features as columns.
W A weight matrix across all observations, i.e inverse of a pairwise distance matrix.

Value

A spatial cross-correlation matrix.

References

Wartenberg, D. Multivariate spatial correlation: A method for exploratory geographical analysis. Geogr. Anal. 17, 263–283 (1985)

Examples

```
{
data.use <- quakes[1:100,]
W <- 1/as.matrix(dist(data.use[,1:2]))
diag(W) <- 0
M <- WartenbergM(data.use[,3:4], W)
}
```

ZPvalue

P-value of Z scores

Description

Calculate p-value of Z scores.

Usage

```
ZPvalue(  
  Z,  
  alternative = c("two.sided", "less", "greater"),  
  p.adjust.method = "BH"  
)
```

Arguments

Z A matrix of Z scores.

alternative Alternative hypothesis used, default is `two.sided`.

p.adjust.method Method used for multiple comparisons correction, default is `BH`. See [p.adjust](#).

Value

A list containing the following:

- `p.val`, a matrix of p-values.
- `p.adj`, a matrix of adjusted p-values.

Examples

```
{  
res <- ZPvalue(matrix(runif(100),10,10))  
}
```

Index

- * **datasets**
 - available.data, 3
- * **data**
 - available.data, 3
 - LoadData, 11
- * **hnn**
 - HnnImpute, 6
 - HnnNeighbor, 7
 - HnnWeight, 8
 - VisiumHnn, 24
- * **local**
 - LocalGi, 12
 - LocalMoransI, 13
 - PermutationGi, 17
 - PermutationLocalI, 18
- * **moransi**
 - BivariateMoransI, 3
 - FitMoransI, 5
 - IrregularData, 8
 - IrregularMoransI, 9
 - OLSMoransI, 15
 - PermutationMoransI, 20
 - UnivariateMoransI, 23
- * **sca**
 - LeeL, 10
 - SCA, 21
 - SpatialXCorr, 22
 - WartenbergM, 25
- * **stats**
 - BivariateMoransIStats, 4
 - LocalMoransIStats, 14
 - PermutationCorr, 16
 - ZPvalue, 26
- _PACKAGE (spots-package), 2

- available.data, 3, 11, 24

- BivariateMoransI, 3, 10
- BivariateMoransIStats, 4

- eigs_sym, 21

- FitMoransI, 5

- HnnImpute, 6
- HnnNeighbor, 7
- HnnWeight, 6, 8

- IrregularData, 8
- IrregularMoransI, 9

- LeeL, 10, 22
- LoadData, 3, 11, 24
- LocalGi, 12
- LocalMoransI, 13
- LocalMoransIStats, 14

- OLSMoransI, 5, 10, 15

- p.adjust, 3, 10, 12, 13, 15, 18, 19, 23, 26
- PermutationCorr, 16
- PermutationGi, 17
- PermutationLocalI, 18
- PermutationMoransI, 20

- SCA, 21
- SpatialXCorr, 21, 22
- spots-package, 2

- UnivariateMoransI, 23

- VisiumHnn, 24

- WartenbergM, 22, 25

- ZPvalue, 26