Package ‘synchrony’

November 9, 2012

Type Package

Title Tools for computing spatial, temporal, and spatiotemporal statistics

Version 0.1

Date 2012-11-06

Author Tarik C. Gouhier

Maintainer Tarik C. Gouhier <tarik.gouhier@gmail.com>

Description Tools for computing spatial, temporal, and spatiotemporal statistics

License GPL (>=2)

URL http://cran.r-project.org/package=synchrony

LazyLoad yes

R topics documented:

synchrony-package *Tools for computing spatial, temporal, and spatiotemporal statistics*

Description

Tools for computing spatial, temporal, and spatiotemporal statistics

Details

Package:	synchrony
Type:	Package
Version:	0.1
Date:	2012-11-06
License:	GPL (>=2)
URL:	http://cran.r-project.org/package=synchrony
LazyLoad:	yes

Author(s)

Tarik C. Gouhier (tarik.gouhier@gmail.com)

Maintainer: Tarik C. Gouhier (tarik.gouhier@gmail.com)

References

Bjornstad, O. N., and W. Falck. 2001. Nonparametric spatial covariance functions: Estimation and testing. *Environmental and Ecological Statistics* 8:53-70.

Bjornstad, O. N., R. A. Ims, and X. Lambin. 1999. Spatial population dynamics: analyzing patterns and processes of population synchrony. *Trends in Ecology & Evolution* 14:427-432.

Buonaccorsi, J. P., J. S. Elkinton, S. R. Evans, and A. M. Liebhold. 2001. Measuring and testing for spatial synchrony. *Ecology* 82:1668-1679.

Cazelles, B., and L. Stone. 2003. Detection of imperfect population synchrony in an uncertain world. *Journal of Animal Ecology* 72:953-968.

Fortin, M. J., and M. R. T. Dale. 2005. Spatial Analysis: A Guide for Ecologists. Cambridge University Press.

Gouhier, T. C., and F. Guichard. 2007. Local disturbance cycles and the maintenance of spatial heterogeneity across scales in marine metapopulations. *Ecology* 88:647-657.

Gouhier, T. C., F. Guichard, and A. Gonzalez. 2010. Synchrony and stability of food webs in metacommunities. *The American Naturalist* 175:E16-E34.

Gouhier, T. C., F. Guichard, and B. A. Menge. 2010. Ecological processes can synchronize marine population dynamics over continental scales. *Proceedings of the National Academy of Sciences* 107:8281-8286.

Loreau, M., and C. de Mazancourt. 2008. Species synchrony and its drivers: Neutral and nonneutral community dynamics in fluctuating environments. *The American Naturalist* 172:E48-E66.

Vasseur, D. A. 2007. Environmental colour intensifies the Moran effect when population dynamics are spatially heterogeneous. *Oikos* 116:1726-1736.

Zar, J. H. 1999. Biostatistical Analysis, Fourth edition. Prentice-Hall, Inc., Upper Saddle River, NJ.

Examples

```
## Compute phase synchrony
t1=cbind(1:100, runif(100))
t2=cbind(1:100, runif(100))
sync=phase.sync(t1, t2)
## Distribution of phase difference
hist(sync$deltaphase$mod_phase_diff_2pi)

## Compute concordant peaks
p=common.peaks.signif(t1, t2)
## Find proportion of time steps where both time series peak together
p$peaks
## Plot (null) distribution of proportion of time steps where both time
## series peak together
hist(p$rand)
## p-value of observed value
p$pval

## Compute Kendall’s W
data(bird.traits)
(w=kendall.w(bird.traits))

## Community matrix for 20 species undergoing random fluctuations
comm.rand=matrix(runif(100), nrow=5, ncol=20)
community.sync(comm.rand)
community.sync.signif(comm.rand, nrands=100)$pval
## Community matrix for 20 species undergoing synchronized fluctuations
comm.corr=matrix(rep(comm.rand[,1], 20), nrow=5, ncol=20)
community.sync(comm.corr)
community.sync.signif(comm.corr, nrands=100)$pval
```

`bird.traits` *bird trait dataset*

Description

Contains the wing length, tail length, and bill length from 12 birds

Usage

```
data(bird.traits)
```

Format

A data frame with 12 observations (birds) on the following 3 variables.

`wing.length` a numeric vector containing wing length in cm

`tail.length` a numeric vector containing tail length in cm

`bill.length` a numeric vector containing bill length in cm

Details

Dataset from Zar (1999; page 444)

Source

Zar, J. H. 1999. Biostatistical Analysis, Fourth edition. Prentice-Hall, Inc., Upper Saddle River, NJ.

Examples

```
data(bird.traits)
(w=kendall.w(bird.traits))
```

`common.peaks.signif` *Find proportion of common peaks/troughs and compute its significance*

Description

Calculates the proportion of local minima/maxima common to both time series and computes its significance by randomly shuffling the time series

Usage

```
common.peaks.signif (t1, t2, nrands = 999)
```

Arguments

`t1` time series 1 in matrix format (`n` rows x 2 columns). The first column should contain the time steps and the second column should contain the values.

`t2` time series 2 in matrix format (`n` rows x 2 columns). The first column should contain the time steps and the second column should contain the values.

`nrands` number of randomizations. Default is `999`.

Value

Returns a list containing:

`pval` p-value computed by randomly shuffling both time series `nrands` times

`rands` proportion of local minima/maxima common to both time series for each randomization

`peaks` proportion of local minima/maxima common to both time series in the observed dataset

`locations` locations/indices of local minima/maxima common to both time series in the observed dataset

Author(s)

Tarik C. Gouhier (tarik.gouhier@gmail.com)

References

Buonaccorsi, J. P., J. S. Elkinton, S. R. Evans, and A. M. Liebhold. 2001. Measuring and testing for spatial synchrony. *Ecology* 82:1668-1679.

See Also

`peaks` and `find.minmax`

Examples

```
t1=cbind(1:100, runif(100))
t2=cbind(1:100, runif(100))
p=common.peaks.signif(t1, t2, nrands=100)
## Find proportion of time steps where both time series peak together
p$peaks
## Plot (null) distribution of proportion of time steps where both time series
## peak together
hist(p$rand)
## p-value of observed value
p$pval
```

`community.sync` *Compute community-wide synchrony*

Description

Compute community-wide synchrony. If all species fluctuate in perfect unison, the community-wide synchrony will be 1. If species undergo uncorrelated fluctuations, the community-wide synchrony will be 0.

Usage

```
community.sync (comm.matrix)
```

Arguments

`comm.matrix`	community matrix in wide format where each row contains the abundance at each time step and each column corresponds to a different species.

Details

Loreau and de Mazancourt (2008) show that community-wide synchrony φ can be quantified by computing the temporal variance $\sigma^2_{x_T}$ of the community time series $x_T(t) = \sum x_i(t)$ and the sum of the temporal standard deviation of the time series across all species $(\sum \sigma_{x_i})^2$ such that: $\varphi = \frac{\sigma^2_{x_T}}{(\sum \sigma_{x_i})^2}$

Value

Returns community-wide synchrony φ

Author(s)

Tarik C. Gouhier (tarik.gouhier@gmail.com)

References

Loreau, M., and C. de Mazancourt. 2008. Species synchrony and its drivers: Neutral and nonneutral community dynamics in fluctuating environments. *The American Naturalist* 172:E48-E66.

Examples

```
## Community matrix for 20 species undergoing random fluctuations
comm.rand=matrix(runif(100), nrow=5, ncol=20)
community.sync(comm.rand)
## Community matrix for 20 species undergoing synchronized fluctuations
comm.corr=matrix(rep(comm.rand[,1], 20), nrow=5, ncol=20)
community.sync(comm.corr)
## Compare this metric to Kendall's W
data(bird.traits)
kendall.w(bird.traits)
community.sync(bird.traits)
```

`community.sync.signif` *Compute significance of community-wide synchrony via Monte Carlo randomizations*

Description

Compute the significance of community-wide synchrony using Monte Carlo randomizations. If all species fluctuate in perfect unison, the community-wide synchrony will be 1. If species undergo uncorrelated fluctuations, the community-wide synchrony will be 0.

Usage

```
community.sync.signif (comm.matrix, nrands = 999)
```

Arguments

`comm.matrix` community matrix in wide format where each row contains the abundance at each time step and each column corresponds to a different species.

`nrands` number of randomizations to perform (default is 999)

Details

Loreau and de Mazancourt (2008) show that community-wide synchrony φ can be quantified by computing the temporal variance $\sigma^2_{x_T}$ of the community time series $x_T(t) = \sum x_i(t)$ and the sum of the temporal standard deviation of the time series across all species $(\sum \sigma_{x_i})^2$ such that: $\varphi = \frac{\sigma^2_{x_T}}{(\sum \sigma_{x_i})^2}$

Value

Returns a named list containing:

`obs`	the observed community synchrony
`rands`	the community synchrony value of the `nrands` randomizations
`pval`	p-value of observed community synchrony

Author(s)

Tarik C. Gouhier (tarik.gouhier@gmail.com)

References

Loreau, M., and C. de Mazancourt. 2008. Species synchrony and its drivers: Neutral and nonneutral community dynamics in fluctuating environments. *The American Naturalist* 172:E48-E66.

See Also

`community.sync`

Examples

```
## Community matrix for 20 species undergoing random fluctuations
comm.rand=matrix(runif(100), nrow=5, ncol=20)
community.sync.signif(comm.rand, nrands=20)$pval
## Community matrix for 20 species undergoing synchronized fluctuations
comm.corr=matrix(rep(comm.rand[,1], 20), nrow=5, ncol=20)
community.sync.signif(comm.corr, nrands=20)$pval
## On "real" data
data(bird.traits)
community.sync.signif(bird.traits, nrands=20)$pval
```

`coord2dist` *coord2dist*

Description

Calculate distance between all pairs of sites

Usage

```
coord2dist (coords, is.latlon = TRUE, lower.tri = TRUE)
```

Arguments

`coords`	n x 4 matrix of coordinates consisting of lat or y, lon or x pairs for each each site
`is.latlon`	are coordinates latitudes/longitudes? Default is `TRUE`
`lower.tri`	Return lower triangular part of the distance matrix? Default is `TRUE`

Value

Returns the distance between all pairs of sites

Author(s)

Tarik C. Gouhier (tarik.gouhier@gmail.com)

Examples

```
coords=rbind(c(32, -125), c(43, -130))
## Compute great circle distance
coord2dist(coords)
```

---

find.minmax *Find min/max of a time series*

Description

Find local minima and maxima of a time series

Usage

```
find.minmax (timeseries)
```

Arguments

timeseries	time series in matrix format (n rows x 2 columns). The first column should contain the time steps and the second column should contain the values.

Value

Returns a list containing:

mins	n x 2 matrix containing the time steps and the local min values
maxs	n x 2 matrix containing the time steps and the local max values

Author(s)

Tarik C. Gouhier (tarik.gouhier@gmail.com)

See Also

find.minmax.ind

Examples

```
t1=cbind(1:100, runif(100))
min.max=find.minmax(t1)
min.max$maxs
plot (t1, t="l")
points (min.max$mins, col="blue", bg="blue", pch=19)
points (min.max$maxs, col="red", bg="red", pch=19)
```

find.minmax.ind *Find indices of local minima or maxima of a time series*

Description

Find indices of local minima or maxima of a time series

Usage

```
find.minmax.ind (timeseries, mins = FALSE)
```

Arguments

`timeseries`	time series in matrix format (n rows x 2 columns). The first column should contain the time steps and the second column should contain the values.
`mins`	find local minima (`mins=TRUE`) or maxima (`mins=FALSE`). Default is `FALSE`.

Value

Returns index of local minima or maxima

Note

This is the low-level function used by `find.minmax`. The optimized version of this function is based on code posted by Ben Bolker on StackOverflow.

Author(s)

Tarik C. Gouhier (tarik.gouhier@gmail.com)

See Also

`find.minmax`

Examples

```
t1=cbind(1:100, runif(100))
## Find local maxima
maxs=find.minmax.ind(t1)
## Find local minima
mins=find.minmax.ind(t1, mins=TRUE)
```

kendall.w *Kendall's W*

Description

Compute Kendall's coefficient of concordance (W)

Usage

```
kendall.w (data)
```

Arguments

data	matrix in wide format where each row represents a different sample and each column represents a different variable.

Details

Kendall's W is a non-parametric statistic that ranges from 0 to 1 and measures the level of agreement between multiple variables. When the number of observations $n > 10$, its significance can be determined by using a χ^2 distribution with $df = n - 1$. Kendall's W is often used to measure the degree of synchrony among different locations.

Value

Returns a named list containing:

w.uncorrected	Kendall's W uncorrected for tied ranks
w.corrected	Kendall's W corrected for tied ranks
pval	p-value of Kendall's W
spearman.corr	Spearman's ranked correlation

Author(s)

Tarik C. Gouhier (tarik.gouhier@gmail.com)

References

Buonaccorsi, J. P., J. S. Elkinton, S. R. Evans, and A. M. Liebhold. 2001. Measuring and testing for spatial synchrony. *Ecology* 82:1668-1679.

Gouhier, T. C., and F. Guichard. 2007. Local disturbance cycles and the maintenance of spatial heterogeneity across scales in marine metapopulations. *Ecology* 88:647-657.

Gouhier, T. C., F. Guichard, and A. Gonzalez. 2010. Synchrony and stability of food webs in metacommunities. *The American Naturalist* 175:E16-E34.

Zar, J. H. 1999. Biostatistical Analysis, Fourth edition. Prentice-Hall, Inc., Upper Saddle River, NJ.

Examples

```
data(bird.traits)
(w=kendall.w(bird.traits))
```

latlon2dist *latlon2dist*

Description

Calculate distance between a pair of coordinates

Usage

```
latlon2dist (coords)
```

Arguments

`coords`	4-element vector of coordinates with format: `(lat1, lon1, lat2, lon2)`

Value

Returns the great circle distance distance between the pair of coordinates

Author(s)

Tarik C. Gouhier (tarik.gouhier@gmail.com)

See Also

`coord2dist`

Examples

```
coords=c(32, -125, 43, -130)
## Compute great circle distance
latlon2dist(coords)
```

peaks *Find common peaks*

Description

Find the proportion of local minima/maxima common to both time series

Usage

```
peaks (t1, t2)
```

Arguments

`t1`	time series 1 in matrix format (`n` rows x 2 columns). The first column should contain the time steps and the second column should contain the values.
`t2`	time series 2 in matrix format (`n` rows x 2 columns). The first column should contain the time steps and the second column should contain the values.

Value

Returns a named list containing peaks and locations

peaks	proportion of local minima/maxima common to both time series
locations	locations/indices of local minima/maxima common to both time series

Note

This is the low-level function used by common.peaks.signif

Author(s)

Tarik C. Gouhier (tarik.gouhier@gmail.com)

References

Buonaccorsi, J. P., J. S. Elkinton, S. R. Evans, and A. M. Liebhold. 2001. Measuring and testing for spatial synchrony. *Ecology* 82:1668-1679.

See Also

common.peaks.signif

Examples

```
t1=cbind(1:100, runif(100))
t2=cbind(1:100, runif(100))
p=peaks(t1, t2)
p$peaks
```

---

phase.partnered *Phase partnered time series*

Description

Create two time series with specific autocorrelation γ and cross-correlation ρ

Usage

```
phase.partnered (n = 2000, rho = 1, gamma = 1, ts.sd = 0.1, ts.mean = 0)
```

Arguments

n	number of time steps in time series. Default is 2000.
rho	cross-correlation between the two time series ($-1 \leq \rho \leq 1$). Default is 1.
gamma	autocorrelation of each time series. Gamma (γ) describes the relationship between frequency f and power P: $P(f) = 1/f^\gamma$ $-2 \leq \gamma \leq 0$: blue noise and $0 \leq \gamma \leq 2$: red noise. Default is 1.
ts.sd	standard deviation of both time series. Default is 0.1.
ts.mean	mean of both time series. Default is 0.

Value

Returns an `n` x 2 matrix containing the time series

Author(s)

Tarik C. Gouhier (tarik.gouhier@gmail.com)

References

Gouhier, T. C., F. Guichard, and A. Gonzalez. 2010. Synchrony and stability of food webs in metacommunities. *The American Naturalist* 175:E16-E34.

Vasseur, D. A. 2007. Environmental colour intensifies the Moran effect when population dynamics are spatially heterogeneous. *Oikos* 116:1726-1736.

Examples

```
## Positively cross-correlated white noise
pos.corr=phase.partnered(n = 100, rho = 0.7, gamma = 0)
## Negatively cross-correlated white noise
neg.corr=phase.partnered(n = 100, rho = -1, gamma = 0)
par(mfrow=c(2,1))
matplot (pos.corr, t="l", lty=1)
matplot (neg.corr, t="l", lty=1)
```

`phase.sync` *Phase synchrony of quasi-periodic time series*

Description

Compute the phase synchrony between two quasi-periodic time series by quantifying their phase difference at each time step

Usage

```
phase.sync (t1, t2, mins = FALSE)
```

Arguments

`t1` time series 1 in matrix format (`n` rows x 2 columns). The first column should contain the time steps and the second column should contain the values.

`t2` time series 2 in matrix format (`n` rows x 2 columns). The first column should contain the time steps and the second column should contain the values.

`mins` use local minima instead of local maxima to compute and then interpolate the phase. Default is `FALSE`.

Details

Two time series are phase-locked if the relationship between their phases remains constant over time. This function computes the phase of successive local maxima or minima for each time series and then uses linear interpolation to find the phase at time steps that fall between local maxima/minima. A histogram can be used to determine if the distribution of the phase difference at each time step is uniform (indicating no phase locking) or has a clear peak (indicating phase locking).

Value

Returns a list containing phases1, phases2, and deltaphase

phases1	n x 3 matrix containing the timestep, value, and phase of the first time series
phases2	n x 3 matrix containing the timestep, value, and phase of the second time series
deltaphase	n x 4 matrix containing the timestep, raw phase difference, phase difference modulus 2π between 0 and 2π, phase difference modulus 2π between $-\pi$ and π

Author(s)

Tarik C. Gouhier (tarik.gouhier@gmail.com)

References

Cazelles, B., and L. Stone. 2003. Detection of imperfect population synchrony in an uncertain world. *Journal of Animal Ecology* 72:953-968.

Examples

```
t1=cbind(1:100, runif(100))
t2=cbind(1:100, runif(100))
## Compute and interpolate phases using successive local minima
sync.mins=phase.sync(t1, t2, mins=TRUE)
## Compute and interpolate phases using successive local maxima
sync.maxs=phase.sync(t1, t2)
## Plot distribution of phase difference
hist(sync.mins$deltaphase$mod_phase_diff_2pi)
```

phase.sync.signif *Compute significance of phase synchrony*

Description

Compute the significance of phase synchrony between two time series via Monte Carlo randomizations

Usage

```
phase.sync.signif (t1, t2, nrands =  999, type = 1, nbreaks = 10, mins = FALSE)
```

Arguments

t1	time series 1 in matrix format (n rows x 2 columns). The first column should contain the time steps and the second column should contain the values.
t2	time series 2 in matrix format (n rows x 2 columns). The first column should contain the time steps and the second column should contain the values.
nrands	number of randomizations to perform (default is 999)
type	flag to indicate whether to compute phase difference modulus 2π between 0 and 2π (type=1) or phase difference modulus 2π between $-\pi$ and π. Default is 1.
nbreaks	number of bins to use to group the values in the time series. Default is 10.
mins	use local minima instead of local maxima to compute and the interpolate the phase. Default is FALSE.

Details

This function determines whether two time series are phase locked by randomly shuffling the time series. Specifically, the distribution of the phase difference of the original time series is binned using a histogram and an index of phase locking Q based on Shannon's entropy is computed: $Q.obs = (S_{max} - S_{obs})/S_{max}$, where $S_{obs} = -\sum_{i}^{N} p_i log(p_i)$, p_i represents the proportion of phase differences in each histogram bin, N the total number of bins, and $S_{max} = log(N)$. $Q.obs = 0$ when there is no phase synchrony and $Q.obs = 1$ when there is perfect phase synchrony. The statistical significance of the phase synchrony $Q.obs$ can then be computed by comparing the observed $Q.obs$ value to the Q values obtained from `nrands` surrogate time series whose short-term correlations and overall temporal patterns match those of the observed time series. The p-value is then the proportion of surrogate time series whose $Q \geq Q.obs$.

Value

Returns a named list containing:

`Q.obs`	Observed Q value
`pval`	p-value of observed `Q.obs`
`rands`	Q values from all randomizations
`phases1`	`n` x 3 matrix containing the timestep, value, and phase of the first time series
`phases2`	`n` x 3 matrix containing the timestep, value, and phase of the second time series
`deltaphase`	`n` x 4 matrix containing the timestep, raw phase difference, phase difference modulus 2π between 0 and 2π, phase difference modulus 2π between $-\pi$ and π
`icdf`	`nrands` x 2 matrix containing the inverse cumulative distribution of the Q values computed using the surrogate time series

Author(s)

Tarik C. Gouhier (tarik.gouhier@gmail.com)

References

Cazelles, B., and L. Stone. 2003. Detection of imperfect population synchrony in an uncertain world. *Journal of Animal Ecology* 72:953-968.

See Also

`phase.sync` and `surrogate.ts`

Examples

```
t1=cbind(1:100, runif(100))
t2=cbind(1:100, runif(100))
p=phase.sync.signif(t1, t2, nrands=99)
p$pval
hist(p$rands)
plot (p$icdf, t="l")
abline(v=p$Q.obs, col="red", lty=2)
```

pisco.data *PISCO multi-year and spatially-explicit mussel and environmental dataset*

Description

Contains the mean annual chl-a concentration, sea surface temperature, upwelling currents, and mussel abundance at 48 intertidal sites along the West Coast of the United States from 2000-2003.

Usage

```
data(pisco.data)
```

Format

A data frame with 192 observations on the following 7 variables.

`latitude` latitude (degrees North)

`longitude` longitude (degrees West)

`chl` mean annual remote sensed chlorophyll-a concentration

`sst` mean annual remote sensed sea surface temperature

`upwelling` mean annual remote sensed upwelling currents

`mussel_abund` mean annual mussel cover (*Mytilus californianus*)

`year` sampling year

References

Gouhier, T. C., F. Guichard, and B. A. Menge. 2010. Ecological processes can synchronize marine population dynamics over continental scales. *Proceedings of the National Academy of Sciences* 107:8281-8286.

Examples

```
data(pisco.data)
```

plot.vario *Plot* `vario` *objects*

Description

Plot `vario` objects

Usage

```
## S3 method for class 'vario'
 plot(x, xlab = "Lag distance", ylab = NULL, ylim = NULL,
                     xtype = c("mean.bin.dist", "bins"), rug = FALSE,
                     pch = 21, col.sig="black", col.nonsig="black", bg.sig="black",
                     bg.nonsig="white", alpha = 0.05, ...)
```

Arguments

`x` vario object generated by vario function.

`xlab` xlabel of the figure. Default is "Lag distance"

`ylab` ylabel of the figure. Default is NULL and will automatically generate the right label

`ylim` y-range. Default is NULL and will automatically generate the best range based on the metric

`xtype` Use either the discrete bin classes (bins) or the mean distance of the points within each bin (mean.bin.dist) on the x-axis. Default is mean.bin.dist

`rug` Plot rug indicating the density of data points? Default is FALSE

`pch` Type of points to use when plotting the variogram. Default is 21

`col.sig` Border color of points for significant values. Default is black

`col.nonsig` Border color of points for non-significant values. Default is black

`bg.sig` Background color of points for significant values. Default is black

`bg.nonsig` Background color of points for non-significant values. Default is black

`alpha` Significance level. Default is 0.05

`...` other graphical parameters.

Author(s)

Tarik C. Gouhier (tarik.gouhier@gmail.com)

Examples

```
data(pisco.data)
d=subset(pisco.data, subset=year==2000, select=c("latitude", "longitude", "sst"))
semiv=vario(data=d)
moran=vario(data=d, type="moran", nrand=100)
geary=vario(data=d, type="geary", nrand=100)

par(mfrow=c(3,1))
plot(semiv)
plot(moran, bg.sig="blue")
plot(geary, bg.sig="red")
```

plot.variofit *Plot* variofit *objects*

Description

Plot variofit objects

Usage

```
## S3 method for class 'variofit'
 plot(x, xlab = "Lag distance", ylab = "Variogram",
                        col.pts = "black", col.line = "red", pch = 21, ...)
```

Arguments

`x`	`variofit` object generated by `vario.fit` function
`xlab`	xlabel of the figure. Default is "Lag distance"
`ylab`	ylabel of the figure. Default is "Variogram"
`col.pts`	Border color of the points. Default is black
`col.line`	Color of the fitted variogram. Default is red
`pch`	Type of points to use when plotting the variogram. Default is 21
`...`	other graphical parameters.

Author(s)

Tarik C. Gouhier (tarik.gouhier@gmail.com)

Examples

```
## Environmental variogram
data(pisco.data)
d=subset(pisco.data, subset=year==2000, select=c("latitude", "longitude", "upwelling"))
semiv=vario(data=d)
mod.sph=vario.fit(semiv$vario, semiv$mean.bin.dist)
plot(mod.sph)
```

`surrogate.ts` *Create surrogate time series via Markov process*

Description

Create surrogate time series with the same short-term time correlation and overall temporal pattern as the original time series using the Markov process described by Cazelles and Stones (2003)

Usage

```
surrogate.ts (ts, distr.ts = NULL, trans.ts = NULL, nbreaks = 10)
```

Arguments

`ts`	time series in matrix format (`n` rows x 2 columns). The first column should contain the time steps and the second column should contain the values.
`distr.ts`	binning of time series values. This parameter must be specified if `trans.ts` is not set to `NULL`. Default is `NULL`.
`trans.ts`	transition matrix from bin i to bin j. Default is `NULL`.
`nbreaks`	number of bins to use to group the time series values. Default is `10`.

Details

The values of the time series x_n are grouped into `nbreak` equally-sized bins. The transition matrix M_{ij} describing the probability of x_{n+1} belonging to bin j based on x_n belonging to bin i is defined using the relative frequencies of the data such that: $M_{ij} = Pr(x_{n+1} \in b_j | x_n \in b_i)$. The surrogate time series is then constructed by randomly selecting a starting value and randomly selecting the next value from the proper bin based on the transition matrix. This process is repeated until the surrogate time series has the same length as the original time series.

Value

Returns a named list containing:

`surr.ts`	surrogate time series in matrix format
`trans`	transition matrix M_{ij}
`distr`	binning of time series values

Author(s)

Tarik C. Gouhier (tarik.gouhier@gmail.com)

References

Cazelles, B., and L. Stone. 2003. Detection of imperfect population synchrony in an uncertain world. *Journal of Animal Ecology* 72:953-968.

See Also

phase.sync.signif

Examples

```
t1=cbind(1:100, runif(100))
surr.t1=surrogate.ts(ts=t1)
plot(t1, t="l")
lines(surr.t1$surr.ts, col="red")
```

`vario` *vario*

Description

Compute the empirical variogram

Usage

```
vario (nbins = 20, extent = 0.5, data, data2 = NULL, is.latlon = TRUE, centered = FALSE,
                  nrands = 0, type = c("semivar", "cov", "pearson",
                  "spearman", "kendall", "moran", "geary"),
                  mult.test.corr = c(FALSE, "holm", "hochberg", "sidak", "bonferroni"))
```

Arguments

`nbins`	Number of bins or lags
`extent`	Proportion of the spatial extent of the data over which to compute the variogram. Default is 0.5 to limit potentially spurious results due to the limited number of data points at large lag distances.
`data`	`n` x `m` matrix containing y-coordinates (or latitude), x-coordinates (or longitude), and values. The values can either be a single column of observations at each site for univariate variograms or a matrix of observations at each site for multivariate variograms (e.g., to compute spatial synchrony).

`data2` `n` x `m` matrix containing y-coordinates (or latitude), x-coordinates (or longitude), and values for second variable. The values can either be a single column of observations at each site for univariate variograms or a matrix of observations at each site for multivariate variograms (e.g., to compute spatial synchrony).

`is.latlon` Are coordinates latitudes/longitudes? Default is `TRUE`

`centered` Should the variogram be centered by subtracting the regional mean from the value in each bin? Default is `FALSE`

`nrands` Number of randomizations to determine statistical significance of variogram. Default is 0.

`type` Type of variogram to compute. Default is `semivar` for semivariance. Other options include `cov` for covariance, `pearson` for Pearson correlation, `spearman` for Spearman correlation, `kendall` for Kendall correlation, `moran` for Moran's I, and `geary` for Geary's C

`mult.test.corr` Correct for multiple tests? Default is `FALSE`. Other options inlcude `holm`, `hochberg`, `sidak`, and `bonferroni`

Details

This function can be used to compute univariate correlograms using Moran's I, Geary's C, and the covariance function or variograms using the semivariance function. Multivariate (Mantel) correlograms can also be computed using the covariance function, Pearson's, Spearman's or Kendall's correlation coefficients. Cross-correlograms/variograms between `data1` and `data2` can be computed with the covariance function, Pearson's, Spearman's or Kendall's correlation coefficients for multivariate variograms and Moran's I, Geary's C, the covariance function, or semivariance for univariate variograms.

Value

Returns a named list containing the following variables:

`bins` Center of each lag/bin

`mean.bin.dist` Mean distance of each lag/bin

`vario` Variogram values in each lag/bin

`npoints` Number of points in each lag/bin

`pvals` p-value for each lag/bin

`rands` `nrands` x `nbins` matrix of randomizations

`metric` Type of variogram computed

`mult.test.corr` Correct for multiple tests?

`is.multivar` Was the analysis performed on multivariate data?

Author(s)

Tarik C. Gouhier (tarik.gouhier@gmail.com)

References

Bjornstad, O. N., and W. Falck. 2001. Nonparametric spatial covariance functions: Estimation and testing. *Environmental and Ecological Statistics* 8:53-70.

Bjornstad, O. N., R. A. Ims, and X. Lambin. 1999. Spatial population dynamics: analyzing patterns and processes of population synchrony. *Trends in Ecology & Evolution* 14:427-432.

Fortin, M. J., and M. R. T. Dale. 2005. Spatial Analysis: A Guide for Ecologists. Cambridge University Press.

See Also

`vario.signif`, `vario.func`

Examples

```
data(pisco.data)
d=subset(pisco.data, subset=year==2000, select=c("latitude", "longitude", "sst"))
semiv=vario(data=d)
moran=vario(data=d, type="moran", nrands=100)
par(mfrow=c(2,1), mar=c(4.2, 4, 1, 1))
plot(semiv$mean.bin.dist, semiv$vario, xlab="Lag distance (km)", ylab="Semivariance")
plot(moran$mean.bin.dist, moran$vario, xlab="Lag distance (km)", ylab="Moran's I", t="l")
points(moran$mean.bin.dist[moran$pvals >= 0.05], moran$vario[moran$pvals >= 0.05],
       bg="white", pch=21)
points(moran$mean.bin.dist[moran$pvals < 0.05], moran$vario[moran$pvals < 0.05],
       bg="black", pch=21)
abline(h=0, lty=2)

## Compute spatial synchrony
d.upw=subset(pisco.data, select=c("latitude", "longitude", "year", "upwelling"))
d.cov=subset(pisco.data, select=c("latitude", "longitude", "year", "mussel_abund"))
## Reshape the data
d.upw.wide=reshape(data=d.upw, idvar=c("latitude", "longitude"), timevar=c("year"),
                   direction="wide")
d.cov.wide=reshape(data=d.cov, idvar=c("latitude", "longitude"), timevar=c("year"),
                   direction="wide")
## Generate variograms
v.upw=vario(nbins=12, data=d.upw.wide, type="pearson", extent=1, nrands=999)
v.cov=vario(nbins=12, data=d.cov.wide, type="pearson", extent=1, nrands=999)
## Fit variograms
v.cov.per=vario.fit(v.cov$vario, v.cov$mean.bin.dist, type="period",
                    start.vals=list(a=1, b=3, c=0))
v.upw.lin=vario.fit(v.upw$vario, v.upw$mean.bin.dist, type="linear")

par(mfrow=c(2,1))
plot(v.cov, xlab="Lag distance (km)", bg.sig="red", col.nonsig="red",
     main="Mussel cover",
     rug=TRUE, ylim=c(-0.3, 0.3))
lines(v.cov$mean.bin.dist, v.cov.per$fit, col="red")
plot(v.upw, xlab="Lag distance (km)", bg.sig="blue", col.nonsig="blue",
     main="Upwelling", rug=TRUE)
lines(v.upw$mean.bin.dist, v.upw.lin$fit, col="blue")
```

vario.fit *vario.fit*

Description

Fit model to the empirical variogram

Usage

```
vario.fit (vario, bins,
                       type = c("spherical", "gaussian", "nugget", "linear",
                       "exponential", "sill", "periodic", "hole"),
                       start.vals = list(c0 = min(vario), c1 = max(vario),
                                         a = max(bins)/4, b=0.1, c=0.1))
```

Arguments

`vario`	Empirical variogram from `emp.vario` function
`bins`	Number of bins or lag distances from `emp.vario` function
`type`	Type of variogram model to fit to the data. Default is `spherical`. Other options are `gaussian`, `nugget`, `linear`, `exponential`, `sill`, `periodic`, and `hole`
`start.vals`	Named list containing the start values for the variogram model: `c0`: nugget, `c1`: sill, `a`: spatial range; `b`: slope; `c`: frequency

Value

Return a named list containing the following variables:

`vario`	Empirical variogram values
`bins`	Empirical variogram bins/lag distances
`AIC`	AIC score of the model fit
`RMSE`	Root Mean Square Error of the model fit
`params`	Named list containing the best estimates for the model parameters
`fit`	Predicted variogram values from the model fit
`nls.success`	did `nls` succeed?

Note

Selecting proper initial values is critical for fitting a reasonable model to the empirical variogram. If these values are off, `nls` will fail and fall-back functions will be used to determine the best parameter values that minimize the Root Mean Square Error (RMSE).

Author(s)

Tarik C. Gouhier (tarik.gouhier@gmail.com)

See Also

`vario.signif`, `vario`, `vario.func`

Examples

```
## Load data
data(pisco.data)
## Environmental variogram
d=subset(pisco.data, subset=year==2000, select=c("latitude", "longitude", "upwelling"))
semiv=vario(data=d)
plot(semiv, xlab="Lag distance (km)")
mod.sph=vario.fit(semiv$vario, semiv$mean.bin.dist)
mod.exp=vario.fit(semiv$vario, semiv$mean.bin.dist, type="expo")
mod.gau=vario.fit(semiv$vario, semiv$mean.bin.dist, type="gauss")
mod.lin=vario.fit(semiv$vario, semiv$mean.bin.dist, type="lin")
lines(semiv$mean.bin.dist, mod.sph$fit, col="red")
lines(semiv$mean.bin.dist, mod.exp$fit, col="black")
lines(semiv$mean.bin.dist, mod.gau$fit, col="blue")
lines(semiv$mean.bin.dist, mod.lin$fit, col="green")
legend(x="topleft", legend=paste(c("Spherical AIC:", "Exponential AIC:",
                                   "Gaussian AIC:", "Linear AIC:"),
                                   c(format(mod.sph$AIC, dig=2),
                                   format(mod.exp$AIC, dig=2),
                                   format(mod.gau$AIC, dig=2),
       format(mod.lin$AIC, dig=2))), lty=1, col=c("red", "black", "blue", "green"),
       bty="n")

## Correlogram
cover=subset(pisco.data, subset=year==2000, select=c("latitude", "longitude", "mussel_abund"))
moran=vario(data=cover, type="moran")
mod.hol=vario.fit(moran$vario, moran$mean.bin.dist, type="hole", start.vals=list(c0=0.6, a=25, c1=0.01))
mod.per=vario.fit(moran$vario, moran$mean.bin.dist, type="period", start.vals=list(a=1, b=3, c=0))
mod.lin=vario.fit(moran$vario, moran$mean.bin.dist, type="linear")
plot(moran, xlab="Lag distance (km)", ylim=c(-0.6, 0.8))
lines(moran$mean.bin.dist, mod.per$fit, col="red")
lines(moran$mean.bin.dist, mod.hol$fit, col="black")
lines(moran$mean.bin.dist, mod.lin$fit, col="blue")
legend(x="topleft", legend=paste(c("Periodic AIC:", "Hole AIC:",
                                   "Linear AIC:"),
                                   c(format(mod.per$AIC, dig=2),
                                   format(mod.hol$AIC, dig=2),
                                   format(mod.lin$AIC, dig=2))),
                                   lty=1, col=c("red", "black", "blue"), bty="n")
```

`vario.func` *vario.func*

Description

Compute the empirical variogram values for each bin

Usage

```
vario.func (x, y, glob.mean, glob.sd, glob.N, is.multivar = FALSE,
                          type=c("semivar", "cov", "pearson",
                                 "spearman", "kendall", "moran", "geary"))
```

Arguments

`x` First set of sites within bin/lag distance

`y` Second set of sites within bin/lag distance

`glob.mean` Global mean

`glob.sd` Global standard deviation

`glob.N` Global number of points

`is.multivar` Is the data multivariate? Default is `FALSE`

`type` Type of variogram to compute. Default is `semivar` for semivariance. Other options include `cov` for covariance, `pearson` for Pearson correlation, `spearman` for Spearman correlation, `kendall` for Kendall correlation, `moran` for Moran's I, and `geary` for Geary's C

Value

Return the value.

Author(s)

Tarik C. Gouhier (tarik.gouhier@gmail.com)

See Also

`vario`, `vario.signif`

Examples

```
## Internal function used by vario
```

vario.signif *vario.signif*

Description

Determine the significance of the empirical variogram via Monte Carlo randomizations

Usage

```
vario.signif (nrands = 999, bins, all.combs, grpdata, data, data2, type,
                              vario, glob.mean = NULL, glob.sd = NULL,
                              glob.N = NULL, is.multivar = FALSE,
                              mult.test.corr=c(FALSE, "holm", "hochberg",
                                               "sidak", "bonferroni"))
```

Arguments

`nrands` Number of randomizations

`bins` Number of bins or lag distances

`all.combs` All combinations of pairs of sites

`grpdata` Grouped sites based on distance bins

`data` `n` x `m` matrix containing y-coordinates (or latitude), x-coordinates (or longitude), and values. The values can either be a single column of observations at each site for univariate variograms or a matrix of observations at each site for multivariate variograms (e.g., to compute spatial synchrony).

`data2` `n` x `m` matrix containing y-coordinates (or latitude), x-coordinates (or longitude), and values. The values can either be a single column of observations at each site for univariate variograms or a matrix of observations at each site for multivariate variograms (e.g., to compute spatial synchrony).

`type` Type of variogram to compute. Default is `semivar` for semivariance. Other options include `cov` for covariance, `pearson` for Pearson correlation, `spearman` for Spearman correlation, `kendall` for Kendall correlation, `moran` for Moran's I, and `geary` for Geary's C

`vario` Empirical variogram

`glob.mean` Global mean

`glob.sd` Global standard deviation

`glob.N` Global number of points

`is.multivar` Is the data multivariate? Default is `FALSE`

`mult.test.corr` Correct for multiple tests? Default is `FALSE`. Other options inlcude `holm`, `hochberg`, `sidak`, and `bonferroni`

Value

Returns a named list containing the following variables:

`rands` `nrands` x `nbins` matrix of randomizations

`pvals` p-values based on one-tailed test

`vario` Variogram values in each lag/bin

`mult.test.corr`
Corrected for multiple tests?

Author(s)

Tarik C. Gouhier (tarik.gouhier@gmail.com)

See Also

`vario`, `vario.func`

Examples

```
## Internal function used by vario
```

Index