

Package ‘wildlifeDI’

February 20, 2015

Type Package

Title Calculate Indices of Dynamic Interaction for Wildlife Telemetry Data

Version 0.2

Date 2014-12-15

Author Jed Long

Maintainer Jed Long <jed.long@st-andrews.ac.uk>

Description Dynamic interaction refers to spatial-temporal associations in the movements of two (or more) animals. This package provides tools for calculating a suite of indices used for quantifying dynamic interaction with wildlife telemetry data. For more information on each of the methods employed see the references within. The package draws heavily on the classes and methods developed in the 'adehabitat' packages.

URL <http://jedalong.github.io/wildlifeDI>

License GPL-3

VignetteBuilder knitr

Suggests knitr, adehabitatHR

Imports sp, rgeos, adehabitatLT

NeedsCompilation no

Repository CRAN

Date/Publication 2014-12-15 15:37:21

R topics documented:

wildlifeDI-package	2
Ca	3
checkTO	4
contacts	5
Cr	6
Cs	7
deer	8

DI	9
Don	11
GetSimultaneous	12
HAI	13
IAB	14
Lixn	16
Prox	18

Index	20
--------------	-----------

wildlifeDI-package	<i>wildlifeDI: Calculate Indices of Dynamic Interaction for Wildlife Telemetry Data</i>
--------------------	---

Description

Dynamic interaction refers to spatial-temporal associations in the movements of two (or more) animals. This package provides tools for calculating a suite of indices used for quantifying dynamic interaction with wildlife telemetry data. For more information on each of the methods employed see the references within. The package draws heavily on the classes and methods developed in the 'adehabitat' packages.

The package wildlifeDI allows users to compute a number of currently available indices of dynamic interaction useful for wildlife telemetry studies. The currently available methods include:

- Prox - Proximity analysis (Bertrand et al. 1996)
- Ca - Coefficient of Association (Bauman 1998)
- Don - Doncaster's measure of dynamic interaction (Doncaster 1990)
- Lixn - Minta's measures of spatial-temporal interaction (Minta 1992)
- Cs - Coefficient of Sociality (Kenward et al. 1993)
- HAI - Half-weight Association Index (Atwood and Weeks Jr. 2003)
- Cr - Correlation coefficient (Shirabe 2006)
- DI - Dynamic interaction index (Long and Nelson 2013)
- IAB - Interaction statistic (Benhamou et al. 2014)

The package wildlifeDI also provides useful functionality for identifying which fixes are temporally simultaneous, required for many of the above methods, using the function GetSimultaneous, along with other functions for exploring spatial-temporal interactions patterns in wildlife telemetry data.

When citing this package please cite the paper Long et al. (2014)

Details

The functions in wildlifeDI utilize the ltraj objects from the package adehabitat (Calenge 2006). For more information on objects of this type see help(ltraj).

Author(s)

Jed Long

References

Calenge, C. (2006) The package "adehabitat" for the R software: A tool for the analysis of space and habitat use by animals. *Ecological Modelling*, **197**: 516-519.

Long, J.A., Nelson, T.A., Webb, S.L., Gee, K.L. (2014) A critical examination of indices of dynamic interaction for wildlife telemetry data. *Journal of Animal Ecology*, **83**: 1216:1233.

(see individual functions for other references)

Ca

*Coefficient of Association***Description**

This function measures the dynamic interaction between two moving objects following the methods first described by Cole (1949), and more recently employed by Bauman (1998).

Usage

```
Ca(traj1, traj2, tc = 0, dc = 50)
```

Arguments

traj1	an object of the class <code>ltraj</code> which contains the time-stamped movement fixes of the first object. Note this object must be a type II <code>ltraj</code> object. For more information on objects of this type see <code>help(ltraj)</code> .
traj2	same as <code>traj1</code> .
tc	temporal tolerance limit (in seconds) for defining when two fixes are simultaneous or together. Parameter passed to function <code>GetSimultaneous</code> .
dc	distance tolerance limit (in appropriate units) for defining when two fixes are spatially together.

Details

This function can be used to calculate the Cole (1949) measure of dynamic interaction between two animals. Termed a coefficient of association, the *Ca* statistic tests the number of fixes the animals are observed together against the total number of fixes following:

$$Ca = \frac{2AB}{A + B}$$

where *A* (respectively *B*) is the number of times animal 1 (resp. 2) are observed, and *AB* is the number of times the two animals are observed together. Several works, including Bauman (1998) have suggested that *Ca* > 0.5 indicates affiliation or fidelity, while *Ca* < 0.5 indicates no association between the two animals. Note that this function calls `GetSimultaneous` to identify the temporal component of identifying when fixes together.

Value

This function returns a numeric result of the Ca statistic.

References

Bauman, P.J. (1998) The Wind Cave National Park elk herd: home ranges, seasonal movements, and alternative control methods. M.S. Thesis. South Dakota State University, Brookings, South Dakota, USA.

Cole, L.C. (1949) The measurement of interspecific association. *Ecology*. **30**, 411–424.

See Also

GetSimultaneous, Prox, HAI

Examples

```
data(deer)
deer37 <- deer[1]
deer38 <- deer[2]
#tc = 7.5 minutes, dc = 50 meters
Ca(deer37, deer38, tc = 7.5*60, dc = 50)
```

checkTO

Check for temporal overlap

Description

The function checkTO is a simple function for identifying if, and for how long, two telemetry datasets overlap (temporally) with each other. The function returns a list with three pieces of information: a logical variable indicating if the two trajectories overlap temporally, and timings of the beginning and end of the overlap period.

Usage

```
checkTO(traj1, traj2)
```

Arguments

traj1	an object of the class <code>ltraj</code> which contains the time-stamped movement fixes of the first object. Note this object must be a type <code>II ltraj</code> object. For more information on objects of this type see <code>help(ltraj)</code> .
traj2	same as traj1.

Details

The function checkTO can be used to identify if, when, and for how long two telemetry datasets overlap temporally.

Value

A list of with three pieces of information, whether the two trajectories overlap ($\$T0$) a logical vector, the beginning ($\$T0start$), and end ($\$T0end$) of the overlap period, stored as POSIX objects.

See Also

GetSimultaneous

Examples

```
data(deer)
deer37 <- deer[1]
deer38 <- deer[2]
spts <- checkT0(deer37, deer38)
```

contacts

Mapping wildlife contacts

Description

The function `contacts` is a simple function for mapping where wildlife contacts occur on the landscape with wildlife telemetry data.

Usage

```
contacts(traj1, traj2, dc = 0, tc = 0,
         proj4string = CRS(as.character(NA)))
```

Arguments

traj1	an object of the class <code>ltraj</code> which contains the time-stamped movement fixes of the first object. Note this object must be a type <code>II ltraj</code> object. For more information on objects of this type see <code>help(ltraj)</code> .
traj2	same as <code>traj1</code> .
tc	time threshold for determining simultaneous fixes – see function: <code>GetSimultaneous</code> .
dc	distance tolerance limit (in appropriate units) for defining when two fixes are spatially together.
proj4string	a string object containing the projection information to be passed included in the output <code>SpatialPolygonsDataFrame</code> object. For more information see the <code>CRS</code> -class in the packages <code>sp</code> and <code>rgdal</code> . Default is <code>NA</code> .

Details

The function `contacts` can be used to map where contacts occur on the landscape, contacts being defined spatially based on a distance threshold `dc` and temporally based on the time threshold `tc` – see the function `getsimultaneous`. The location of the contact is calculated as the midpoint between the two fixes that are determined to be a "contact" based on `dc` and `tc`.

Value

A `SpatialPointsDataFrame` containing the locations of the contacts. The time of the contact is stored in the attributes of the `SpatialPointsDataFrame` object, along with the actual distance between fixes.

See Also

`GetSimultaneous`, `Prox`

Examples

```
data(deer)
deer37 <- deer[1]
deer38 <- deer[2]
#tc = 7.5 minutes, dc = 50 meters
spts <- contacts(deer37, deer38, tc=7.5*60, dc=50)
```

Cr *Movement Correlation Coefficient*

Description

The function `Cr` computes the correlation statistic for movement data as presented in the paper by Shirabe (2006). The statistic is essentially a Pearson product-moment correlation statistic formulated for use with movement data.

Usage

```
Cr(traj1, traj2, tc = 0)
```

Arguments

<code>traj1</code>	an object of the class <code>ltraj</code> which contains the time-stamped movement fixes of the first object. Note this object must be a type II <code>ltraj</code> object. For more information on objects of this type see <code>help(ltraj)</code> .
<code>traj2</code>	same as <code>traj1</code> .
<code>tc</code>	time threshold for determining simultaneous fixes – see function: <code>GetSimultaneous</code> .

Details

The function `Cr` can be used to measure the level of dynamic interaction (termed correlation) between a pair of simultaneously moving objects. The statistic is sensitive to interaction in both movement direction (azimuth) and displacement, but is unable to disentangle the effects of these components.

Value

This function returns the Shirabe (2006) correlation statistic for two moving objects.

References

Shirabe, T. 2006. Correlation analysis of discrete motions. In: Raubal, M., Miller, HJ, Frank, AU, and Goodchild, M. eds. GIScience 2006, LNCS 4197. Berlin: Springer-Verlag; 370-382.

See Also

GetSimultaneous, DI

Examples

```
data(deer)
deer37 <- deer[1]
deer38 <- deer[2]
#tc = 7.5 minutes
Cr(deer37, deer38, tc = 7.5*60)
```

Cs

Coefficient of Sociality

Description

The function Cs computes the coefficient of sociality between two moving objects following the methods outlined by Kenward et al. (1993). It also uses a signed Wilcoxon-rank test to test for significance.

Usage

```
Cs(traj1, traj2, tc = 0)
```

Arguments

traj1	an object of the class <code>ltraj</code> which contains the time-stamped movement fixes of the first object. Note this object must be a type II <code>ltraj</code> object. For more information on objects of this type see <code>help(ltraj)</code> .
traj2	same as traj1.
tc	time threshold for determining simultaneous fixes – see function: <code>GetSimultaneous</code> .

Details

This function can be used to calculate the Kenward et al. (1993) coefficient of sociality (Cs) between two animals. The Cs statistic tests the observed mean distance between simultaneous fixes against that expected by the overall distribution of distances between all fixes.

$$C_s = \frac{D_E - D_O}{D_O + D_E}$$

Where D_O is the mean observed distance between simultaneous fixes, and D_E is the mean expected distance between all fixes. Kenward et al. (1993) propose Cs as a useful metric for exploring attraction or avoidance behaviour. Values for Cs closer to 1 indicate attraction, while values for Cs

closer to -1 indicate avoidance. Values of Cs near 0 indicate that the two animals' movements have no influence on one another.

Further, the difference between the observed and expected distances are compared using a paired signed-rank test (both one-sided tests, indicative of attraction or avoidance). See the function `GetSimultaneous` for details on how simultaneous fixes are determined from two trajectories.

Value

This function returns a list of objects representing the calculated values from the Cs statistic and associated p -values from the signed rank test.

- Do – The mean distance of simultaneous fixes.
- De – The mean expected distance, from all fixes.
- Cs – The coefficient of sociality, see **Details**.
- p.Attract – One sided p -value from signed rank test, testing for attraction.
- p.Avoid – One sided p -value from signed rank test, testing for avoidance.

References

Kenward, R.E., Marcstrom, V. and Karlbom, M. (1993) Post-nestling behaviour in goshawks, *Accipiter gentilis*: II. Sex differences in sociality and nest-switching. *Animal Behaviour*. **46**, 371–378.

See Also

`GetSimultaneous`

Examples

```
data(deer)
deer37 <- deer[1]
deer38 <- deer[2]
#tc = 7.5 minutes
Cs(deer37, deer38, tc = 7.5*60)
```

deer

GPS tracking data of two male deer

Description

GPS telemetry data for two male deer during a one-week period in March 2005. The two deer form a male bachelor group, making them an interesting case study for studying dynamic interaction patterns. The data are a subset of the data used as a case study in Long *et al.* (2014).

Format

An `ltraj` object with two bursts, representing the two different individual deer:

- Deer no. 37 containing 551 fixes.
- Deer no. 38 containing 567 fixes.

Details

The deer data are stored as a single `'ltraj'` object; two bursts contain the fixes for two individuals (deer37 and deer 38). GPS fixes were attempted at a regular sampling frequency of 15 minutes. For more information on these data how the deer data was collected or for citation please see the papers Webb *et al.* (2009, 2010).

References

Long, J.A., Nelson, T.A., Webb, S.L., Gee, K.L. (2014) A critical examination of indices of dynamic interaction for wildlife telemetry studies. *Journal of Animal Ecology*, **83**: 1216-1233.

Webb, S.L., Gee, K.L., Demarais, S., Strickland, B.K., DeYoung, R.W. (2009) Efficacy of a 15-strand high-tensile electric fence to control white-tailed deer movements. *Wildlife Biology in Practice*, **5**, 45-57.

Webb, S.L., Gee, K.L., Strickland, B.K., Demarais, S., DeYoung, R.W. (2010) Measuring fine-scale white-tailed deer movements and environmental influences using GPS collars. *International Journal of Ecology*, **2010**, 1-12.

Examples

```
data(deer)
deer37 <- deer[1]
deer38 <- deer[2]
plot(deer37)
plot(deer38)
```

DI

Dynamic interaction index

Description

The function `DI` measures dynamic interaction between two moving objects. It calculates the local level `di` statistic for movement displacement, direction, and overall. `DI` can compute time- and/or distance-based weighting schemes following Long and Nelson (2013).

Usage

```
DI(traj1, traj2, tc = 0, alpha = 1, local = FALSE, TimeWeight = FALSE,
   DistWeight = FALSE)
```

Arguments

traj1	an object of the class <code>ltraj</code> which contains the time-stamped movement fixes of the first object. Note this object must be a type <code>II</code> <code>ltraj</code> object. For more information on objects of this type see <code>help(ltraj)</code> .
traj2	same as <code>traj1</code> .
tc	time threshold for determining simultaneous fixes – see function: <code>GetSimultaneous</code> .
alpha	value for the α parameter in the formula for di_d .
TimeWeight	whether or not to compute the weights for time-based weighting.
DistWeight	whether or not to compute the weights for distance-based weighting.
local	logical, whether or not to perform local analysis.

Details

This function can be used for calculating the dynamic interaction (DI) statistic as described in Long and Nelson (2013). The DI statistic can be used to measure the local level of dynamic interaction between two moving objects. Specifically, it measures dynamic interaction in movement direction and displacement.

Value

The function `DI` returns the DI index (along with DI_{theta} and DI_d), or if `local = TRUE`, a dataframe that contains the localized `di` values (see Long and Nelson 2013). The columns for `di`, `di.theta`, and `di.d` represent dynamic interaction overall, in direction (azimuth), and in displacement respectively. If time- and/or distance-based weighting is selected, the corresponding weights are included as additional columns `t.weight` and `d.weight`, respectively. Please see Long and Nelson (2013) for a more detailed description.

References

Long, J.A., Nelson, T.A. 2013. Measuring dynamic interaction in movement data. *Transactions in GIS*. 17(1): 62-77.

See Also

`GetSimultaneous`, `Cr`

Examples

```
data(deer)
deer37 <- deer[1]
deer38 <- deer[2]
#tc = 7.5 minutes
DI(deer37, deer38, tc = 7.5*60)
df <- DI(deer37, deer38, tc = 7.5*60, local = TRUE)
```

Don

Doncaster's measure of dynamic interaction

Description

The function Don measures the dynamic interaction between two moving objects following the methods outlined by Doncaster (1990).

Usage

```
Don(traj1, traj2, tc = 0, dc = 50)
```

Arguments

traj1	an object of the class ltraj which contains the time-stamped movement fixes of the first object. Note this object must be a type II ltraj object. For more information on objects of this type see <code>help(ltraj)</code> .
traj2	same as traj1.
tc	time threshold for determining simultaneous fixes – see function: <code>GetSimultaneous</code> .
dc	distance tolerance limit (in appropriate units) for defining when two fixes are spatially together.

Details

This function can be used to compute the Doncaster (1990) methods for measuring dynamic interaction between two objects. The Doncaster method tests the proportion of simultaneous fixes that are below dc against that which would be expected based on the distribution of distances between all fixes.

Value

This function first returns a plot, for distance values ranging from 0 to the maximum distance separating two fixes, of the observed proportion of simultaneous fixes below each distance value. The expected values based on all fixes are also included. Second, a list is returned that contains the contingency table of simultaneous fixes (paired) and non-paired fixes below and above dc, along with the associated *p*-value from the Chi-squared test.

- conTable – contingency table showing frequency of paired and non-paired fixes above and below dc.
- p.value – *p*-value from the Chi-squared test of conTable.

References

Doncaster, C.P. (1992) Non-parametric estimates of interaction from radio-tracking data. *Journal of Theoretical Biology*, **143**: 431-443.

See Also

GetSimultaneous

Examples

```
data(deer)
deer37 <- deer[1]
deer38 <- deer[2]
#tc = 7.5 minutes, dc = 50 meters
Don(deer37, deer38, tc = 7.5*60, dc = 50)
```

GetSimultaneous

Identify simultaneous fixes between trajectories

Description

The function `GetSimultaneous` identifies and extracts simultaneous fixes, within a given tolerance limit, between two movement datasets.

Usage

```
GetSimultaneous(traj1, traj2, tc = 0)
```

Arguments

<code>traj1</code>	an object of the class <code>ltraj</code> which contains the time-stamped movement fixes of the first object. Note this object must be a type <code>II</code> <code>ltraj</code> object. For more information on objects of this type see <code>help(ltraj)</code> .
<code>traj2</code>	same as <code>traj1</code> .
<code>tc</code>	time threshold for determining simultaneous fixes. For simplicity, <code>tc</code> is always taken in seconds.

Details

This function is used to determine the simultaneous fixes between two movement datasets facilitating further analysis.

Value

A single `ltraj` object containing two bursts, representing the two original `ltraj` objects, each containing only those fixes that are deemed simultaneous.

See Also

`as.ltraj`

Examples

```

data(deer)
deer37 <- deer[1]
deer38 <- deer[2]
#tc = 7.5 minutes
trajs <- GetSimultaneous(deer37, deer38, tc = 7.5*60)
deer37 <- trajs[1]
deer38 <- trajs[2]

```

HAI

*Half-weight Association Index***Description**

This function computes the Half-weight Association Index for examining the presence of dynamic interaction in wildlife telemetry studies. This implementation follows that outlined in the paper Atwood and Weeks (2003).

Usage

```
HAI(traj1, traj2, OZ, tc = 0, dc = 50)
```

Arguments

traj1	an object of the class <code>ltraj</code> which contains the time-stamped movement fixes of the first object. Note this object must be a type II <code>ltraj</code> object. For more information on objects of this type see <code>help(ltraj)</code> .
traj2	same as <code>traj1</code> .
OZ	spatial polygon associated with the home range (or some other form of) spatial overlap between <code>traj1</code> and <code>traj2</code> . Required to be an object that coerces to class <code>SpatialPolygons</code> .
tc	time threshold for determining simultaneous fixes – see function: <code>GetSimultaneous</code> .
dc	distance tolerance limit (in appropriate units) for defining when two fixes are spatially together.

Details

This function can be used to test for the presence of dynamic interaction within the shared area (often termed the overlap zone) of the two animals home ranges. Specifically, HAI is calculated in identical fashion to that for `Ca`, but considers only those fixes in the shared area. Typically, the overlap zone (OZ) is easily obtained by taking the spatial intersection of two polygon home ranges.

Value

This function returns the numeric value of the HAI statistic. Values near 1 indicate attraction within the shared home range area, while values near 0 indicate avoidance within this shared area.

References

Atwood, T.C. and Weeks Jr., H.P. (2003) Spatial home-range overlap and temporal interaction in eastern coyotes: The influence of pair types and fragmentation. *Canadian Journal of Zoology*, **81**: 1589-1597.

See Also

GetSimultaneous, Ca

Examples

```
## Not run:
data(deer)
deer37 <- deer[1]
deer38 <- deer[2]
library(adehabitatHR)
library(sp)
library(rgeos)
#use minimum convex polygon for demonstration...
hr37 <- mcp(SpatialPoints(ld(deer37)[,1:2]))
hr38 <- mcp(SpatialPoints(ld(deer38)[,1:2]))
OZ <- gIntersection(hr37,hr38)
#tc = 7.5 minutes, dc = 50 meters
HAI(deer37, deer38, OZ=OZ, tc=7.5*60, dc=50)

## End(Not run)
```

IAB

Benhamou's IAB Index

Description

The function IAB computes the IAB index following the methods described in the paper by Benhamou et al. (2014). It facilitates global analysis, with the significance testing procedure described in the paper, but also a local level output, to explore the IAB statistic through time.

Usage

```
IAB(traj1, traj2, tc = 0, dc = 50, local = FALSE)
```

Arguments

traj1	an object of the class <code>ltraj</code> which contains the time-stamped movement fixes of the first object. Note this object must be a type <code>II</code> <code>ltraj</code> object. For more information on objects of this type see <code>help(ltraj)</code> .
traj2	same as <code>traj1</code> .

tc	time threshold for determining simultaneous fixes – see function: <code>GetSimultaneous</code> .
dc	critical distance where the IAB function will show maximum slope – see Benhamou et al. (2014) for more advice on selecting this parameter.
local	logical value indicating whether a dataframe (<code>local = TRUE</code>) containing the IAB index for each simultaneous fix should be returned, or (if <code>local = FALSE</code> - the default) the global index along with associated p-value.

Details

The function IAB can be used to test for direct interaction in wildlife telemetry data and affords a novel significance testing procedure that takes into account the serially correlated structure of telemetry data. Specifically, it computes an index analogous to the Bhattacharyya coefficient between the potential influence domains of two animals. Like the other indices, IAB is dependent on the selection of an appropriate value for dc (which is termed Δ in the article). The dc parameter here is not a threshold distance, but rather the distance at which the function shows maximum slope (see Benhamou et al. 2014).

The significance testing procedure uses a wrapped shifting method in order to maintain the serially correlated structure of the data. At each shift, a sample value of IAB (termed MAB) is computed in order to generate a distribution of values to test against (for more information see Benhamou et al. 2014).

Value

If `local=FALSE` (the default) IAB returns the numeric value of the IAB index and the associated p-value. If `local=TRUE` IAB returns a dataframe (containing the date/times of *all* simultaneous fixes, along with the distance between fixes at each time, and the IAB index value for each simultaneous fix).

References

Benhamou, S., Valeix, M., Chamaille-Jammes, S., Macdonald, D., Loveridge, A.J. (2014) Movement-based analysis of interactions in African lions. *Animal Behaviour*, **90**: 171-180.

See Also

`GetSimultaneous`, `DI`, `Prox`

Examples

```
data(deer)
deer37 <- deer[1]
deer38 <- deer[2]
#tc = 7.5 minutes, dc = 50 meters
IAB(deer37, deer38, tc=7.5*60, dc=50)
df <- IAB(deer37, deer38, tc=7.5*60, dc=50, local=TRUE)
```

Lixn

*Minta's Spatial-temporal interaction statistics***Description**

The function `Lixn` measures dynamic interaction between two animals following the methods outlined by Minta (1992).

Usage

```
Lixn(traj1, traj2, method = "spatial", tc = 0, hr1, hr2, OZ = NULL)
```

Arguments

<code>traj1</code>	an object of the class <code>ltraj</code> which contains the time-stamped movement fixes of the first object. Note this object must be a type <code>II ltraj</code> object. For more information on objects of this type see <code>help(ltraj)</code> .
<code>traj2</code>	same as <code>traj1</code> .
<code>method</code>	method for computing the marginal distribution from which expected values are computed. If <code>method = "spatial"</code> , the marginal values are calculated based on areas of the shared and unshared portions of the home ranges. If <code>method = "frequency"</code> , the marginal values are calculated based on the number of all fixes within the shared and unshared portions of the home ranges – see <code>Details</code> .
<code>tc</code>	time threshold for determining simultaneous fixes – see function: <code>GetSimultaneous</code> .
<code>hr1</code>	(– required if <code>method = 'spatial'</code>) home range polygon associated with <code>traj1</code> . Must be an object that coerces to class <code>SpatialPolygons*</code> .
<code>hr2</code>	(– required if <code>method = 'spatial'</code>) same as <code>hr1</code> , but for <code>traj2</code> .
<code>OZ</code>	(– required if <code>method = 'frequency'</code>) shared area polygon associated with spatial use overlap between <code>traj1</code> and <code>traj2</code> . Must be an object that coerces to class <code>SpatialPolygons*</code> .

Details

The function `Lixn` can be used to calculate the Minta (1992) measures of dynamic interaction between two animals. The Minta statistic tests how the two animals simultaneously utilize an area shared between the two individuals. Three coefficients are produced L_{AA} , L_{BB} , and L_{ixn} . Each of these statistics are based on a contingency table that compares the observed frequency of those fixes that are simultaneous and within/outside the shared area to expectations based on area overlap proportions (if `method="spatial"`) or expectations derived from all fixes (if `method="frequency"`) – see Minta (1992) for more details. A Chi-squared statistic can then be used to examine the significance between the observed and expected use of the shared area.

Minta (1992) suggests the following interpretations of the coefficients. When L_{AA} is near 0, the first animal's use of the shared area is random (or as expected). When $L_{AA} > 0$ it signifies spatial

attraction to the shared area, or greater than expected use. When $L_{AA} < 0$ it signifies spatial avoidance of the shared area, or less than expected use. Interpretation of L_{BB} is the same as for L_{AA} with respect to the second animal. L_{ixn} tells us far more about the nature of the interaction between the two individuals. As L_{ixn} nears 0, both animals use the shared area randomly, with regards to the other animal. If $L_{ixn} > 0$ the animals use the shared area more *simultaneously*, whereas if $L_{ixn} < 0$ it is an indication of *solitary* use, or avoidance. This is why L_{ixn} is termed the temporal interaction coefficient. A Chi-squared test can be used to identify the significance of the L_{AA} , L_{BB} , and L_{ixn} values.

NOTES:

1. With modern telemetry datasets, where home ranges are readily estimated, choosing method = 'spatial' is most appropriate.
2. When the home ranges do not overlap the Lixn statistic is not defined and the function returns a string of NA's.
3. When one home range completely encloses another the Lixn statistic is not defined and the function returns a string of NA's and 'ContainsB' (or 'ContainsB') under the p.IXN result.
4. Further to points 2 and 3, the Lixn statistic is not appropriate in situations where the overlap area is either very large or very small relative to either home range (i.e., a situation with almost complete enclosure or virtually no overlap). Thus, it is advised that Lixn be used only in situations where there are suitable marginal areas for areaA, areaB, and areaAB – see Minta (1992).

Value

This function returns a list of objects representing the calculated values from the Minta statistic and associated p -values from the Chi-squared test.

- pTable – contingency table showing marginal probabilities of expected use based on the selection of the method parameter.
- nTable – contingency table showing observed frequency of use of the shared area based on simultaneous fixes.
- oTable – the odds for each cell in the contingency table.
- Laa – the calculated value of the L_{AA} statistic
- p.AA – the associated p -value
- Lbb – the calculated value of the L_{BB} statistic
- p.BB – the associated p -value
- Lixn – the calculated value of the L_{ixn} statistic
- p.IXN – the associated p -value

References

Minta, S.C. (1992) Tests of spatial and temporal interaction among animals. *Ecological Applications*, **2**: 178-188

See Also

GetSimultaneous

Examples

```
## Not run:
data(deer)
deer37 <- deer[1]
deer38 <- deer[2]
library(adehabitatHR)
library(sp)
#use minimum convex polygon for demonstration...
hr37 <- mcp(SpatialPoints(ld(deer37)[,1:2]))
hr38 <- mcp(SpatialPoints(ld(deer38)[,1:2]))
#tc = 7.5 minutes, dc = 50 meters
Lixn(deer37, deer38, method='spatial', tc=7.5*60, hr1=hr37, hr2=hr38)

## End(Not run)
```

Prox

Proximity Index

Description

The function Prox simply computes the proportion of fixes that are proximal, based on some spatial threshold – dc (Bertrand et al. 1996). It also facilitates local-level proximity analysis

Usage

```
Prox(traj1, traj2, tc = 0, dc = 50, local = FALSE)
```

Arguments

traj1	an object of the class <code>ltraj</code> which contains the time-stamped movement fixes of the first object. Note this object must be a type II <code>ltraj</code> object. For more information on objects of this type see <code>help(ltraj)</code> .
traj2	same as traj1.
tc	time threshold for determining simultaneous fixes – see function: <code>GetSimultaneous</code> .
dc	distance tolerance limit (in appropriate units) for defining when two fixes are spatially together.
local	logical value indicating whether or not a dataframe, containing the distance between each simultaneous fix, should be returned.

Details

The function Prox can be used to test for the presence of attraction (via proximity) in wildlife telemetry data. Prox is simply the proportion of simultaneous fixes within the threshold distance – dc. The local output (dataframe) can be useful for examining variation in proximity through time.

Value

If `local=FALSE` (the default) `Prox` returns the numeric value of the Prox index. If `local=TRUE` `Prox` returns a dataframe (containing the date/times of *all* simultaneous fixes, along with the distance between fixes at each time.

References

Bertrand, M.R., DeNicola, A.J., Beissinger, S.R., Swihart, R.K. (1996) Effects of parturition on home ranges and social affiliations of female white-tailed deer. *Journal of Wildlife Management*, **60**: 899-909.

See Also

`GetSimultaneous`, `DI`, `contacts`

Examples

```
data(deer)
deer37 <- deer[1]
deer38 <- deer[2]
#tc = 7.5 minutes, dc = 50 meters
Prox(deer37, deer38, tc=7.5*60, dc=50)
df <- Prox(deer37, deer38, tc=7.5*60, dc=50, local=TRUE)
```

Index

*Topic **datasets**

deer, 8

*Topic **indices**

Ca, 3

contacts, 5

Cr, 6

Cs, 7

DI, 9

Don, 11

HAI, 13

IAB, 14

Lixn, 16

Prox, 18

*Topic **overlap**

checkT0, 4

*Topic **processing**

GetSimultaneous, 12

*Topic **temporal**

checkT0, 4

Ca, 3

checkT0, 4

contacts, 5

Cr, 6

Cs, 7

deer, 8

DI, 9

Don, 11

GetSimultaneous, 12

HAI, 13

IAB, 14

Lixn, 16

Prox, 18

wildlifeDI-package, 2