wildlifeTG

December 22, 2014

Type Package

Title Time Geograhic Analysis of Wildlife Telemetry Data

Version 0.2

Date 2014-12-22

Author Jed Long

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

Description The package wildlifeTG provides tools for performing time geographic analysis of animal tracking data. The functions provide useful tools for examining wildflife movement from the context of accessibility. The time geographic framework is an alternative view on typical home range estimation procedures. Currently, the package provides functions that faciliate the calculation of the PPA and dynPPA measures of an individuals accessibility space. Please note that the package is still under development. Please see the wildlifeTG website at http://jedalong.github.io/wildlifeTG for the latest information and most up-to-date version of the package.

URL http://jedalong.github.io/wildlifeTG

License GPL-3

Imports sp,
adehabitatLT,
rgeos

R topics documented:

wildlifeTG-package *wildlifeTG - Time Geographic Analysis of Wildlife Telemetry Data*

Description

The package wildlifeTG provides tools for performing time geographic analysis of animal tracking data. The functions provide useful tools for examining wildflife movement from the context of accessibility. The time geographic framework is an alternative view on typical home range estimation procedures. Currently, the package provides functions that faciliate the calculation of the PPA and dynPPA measures of an individuals accessibility space. Please note that the package is still under development. Please see the wildlifeTG website at http://jedalong.github.io/wildlifeTG for the latest information and most up-to-date version of the package.

Details

wildlifeTG's functions utilize the ltraj objects from the package adehabitat.

Author(s)

Jed Long

References

Long, JA, Nelson, TA. (2012) Time geography and wildlife home range delineation. *Journal of Wildlife Management*, 76(2):407-413.

Long, JA, Nelson, TA. (2014) Home range and habitat analysis using dynamic time geography. *Journal of Wildlife Management*. Accepted: 2014-12-03.

dynppa *Dynamic PPA Measure of Animal Space Use*

Description

The function dynppa computes the (dynamic) PPA measure of the accessibility space of an animal. The PPA method can be thought as an alternative view on the home range; one that explicitly considers the spatial and temporal constraints on movement given known telemetry fixes, and a (dynamic) measure of maximum mobility - termed Vmax. The PPA method incorporates dynamic behaviour into the calculation of the vmax parameter used to delineate the original version of the PPA method, but the original method is still an option here.

Usage

```
dynppa(traj, tol = max(ld(traj)$dt, na.rm = TRUE), dissolve = TRUE,
  proj4string = CRS(as.character(NA)), ePoints = 360, ...)
```

Arguments

- `traj` — 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 `help(ltraj)`.
- `tol` — parameter used to filter out those segments where the time between fixes is overly large (often due to irregular sampling or missing fixes); which leads to an overestimation of the activity space via the PPA method. Default is the maximum sampling interval from `traj`.
- `dissolve` — (logical) whether or not to dissolve output elliplse polygons to create a single output polygon, or keep the individual segment PPA ellipses. Default = TRUE.
- `proj4string` — a string object containing the projection information to be passed included in the output `SpatialPolygonsDataFrame` object. For more information see the `CRS-class` in the packages `sp` and `rgdal`. Default is `NA`.
- `ePoints` — number of vertices used to construct each PPA ellipse. More points will necessarily provide a more detailed ellipse shape, but will slow computation; default is 360.
- `...` — additional parameters to be passed to the function `dynvmax`. For example, should include `method` and/or `dynamic` parameters, see the documentation for `dynvmax` for more detailed information on what to include here.

Details

The function `dyn.ppa` represents an extension to an existing PPA method (Long and Nelson, 2012). Dynamic calculation of the PPA method improves upon the original version by flexibly modelling the vmax parameter according to wildlife behaviour. See the function `dyn.vmax` for more information on how to incorporate dynamic behaviour into the vmax parameter estimation.

Value

This function returns a `SpatialPolygonsDataFrame` representing the dynamic PPA measure of the accessibility space of an individual animal.

References

Long, JA, Nelson, TA. (2012) Time geography and wildlife home range delineation. *Journal of Wildlife Management*, 76(2):407-413.

Long, JA, Nelson, TA. (2014) Home range and habitat analysis using dynamic time geography. *Journal of Wildlife Management*. Accepted: 2014-12-03.

See Also

dynvmax

dynvmax *Dynamic Calculation of the Vmax Parameter*

Description

The function dynvmax computes a dynamic version of the Vmax parameter for the PPA method. It can be used to incorporate changes in animal movement behaviour into the PPA method calculation to better model that area accessible to an individual animal given the set of known telemetry locations in space and time.

Usage

```
dynvmax(traj, dynamic = "NA", w = 9, class.col = "dt",
  method = "Robson", k = 5, alpha = 0.05, manualVmax = NA,
  vmaxtrunc = NA)
```

Arguments

traj	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 help(ltraj).
dynamic	one of 'NA', 'focal', 'cumulative', or 'class'; which signifies whether or how to dynamically compute the Vmax parameter. See **Details** for more information on each of the choices.
method	method for computing the Vmax parameter dynamically; can be one of several options: – "Robson" for the Robson & Whitlock (1964) method, – "RobsonLL" for the R & W (1964) lower $(1-\alpha)*100\%$ C.I. limit, – "RobsonUL" for the R & W (1964) upper $(1-\alpha)*100\%$ C.I. limit, – "vanderWatt" for the van der Watt (1980) method, – "vanderWattLL" for the van der Watt (1980) lower $(1-\alpha)*100\%$ C.I. limit, – "vanderWattUL" for the van der Watt (1980) upper $(1-\alpha)*100\%$ C.I. limit.
w	(optional) window size (only used with dynamic = 'focal' or 'cumulative').
class.col	(optional) character indicating the name of the column in the infolocs dataframe of traj containing the categorized behavioural states of the animal (which can be stored as a character or numeric column).
k	(optional) value for the *k* parameter in the van der watt (1980) method; default is 5.
alpha	(optional) value for the α parameter if using upper or lower C.I. methods; default is 0.05.
manualVmax	(optional) Character name of column in traj storing user input column of vmax values (typically call the column dynVmax).
vmaxtrunc	(optional) due to irregular sampling intervals, or errors in GPS location, or other effects, the calculation of the vmax parameter through the statistical methods outlined above can be heavily influenced by high outliers. Thus, it may be useful to exclude those segments from calculation of the dynamic Vmax parameter. Default is NA.

Details

The function dynvmax represents an intermediary function used to extend and improve upon an existing PPA home range method (Long and Nelson, 2012) as described in the paper (Long and Nelson, 2014). Four options are available for computing the vmax parameter dynamically and are passed into the dynvmax function using dynamic option.

1) NA – if dynamic = 'NA' (the default) the function estimates the original, non-dynamic estimate of Vmax which is a global estimate, as per Long & Nelson (2012).
2) focal – a moving window approach whereby a window of size w is moved along the trajectory and vmax computed dynamically within each window and assigned to the central segment.
3) cumulative – A moving window of size w is again used, only in this case the value is assigned to the end segment. This represents the vmax calculation of the previous w segments.
4) class – A priori analysis (e.g., obtained via state-space models, or from expert knowledge) is used to identify discrete behavioural states in the telemetry data and these stored in a column which is then passed into the function.

The class method is the preferred choice, as it allows the use of more sophisticated models for identifying behavioural shifts in telemetry data where we would expect to see clear differences in the Vmax parameter based on changing movement behaviour.

The use of the 'focal' or 'cumulative' dynamic methods uses a moving window approach, which is sensitive to edge effects at the initial and ending times of the trajectory. Thus, the dynamic Vmax parameter is only computed for those segments that have a valid window and the dataset is shrunk by w-1 segments.

Value

This function returns the original traj object with a new column – dynVmax in the infolocs dataframe containing the dynamic vmax parameter for each trajectory segment.

References

Long, JA, Nelson, TA. (2012) Time geography and wildlife home range delineation. *Journal of Wildlife Management*. 76(2):407-413.

Long, JA, Nelson, TA. (2014) Home range and habitat analysis using dynamic time geography. *Journal of Wildlife Management*. Accepted: 2014-12-03.

Robson, DS, Whitlock, JH. (1964) Estimation of a truncation point. *Biometrika* 51:33-39.

van der Watt, P. (1980) A note on estimation bounds of random variables. *Biometrika* 67(3):712-714.

See Also

dynppa

`jppa` *Joint Potential Path Area of Two Animals*

Description

The function `jppa` computes the joint accessibility space between two animals. It can be used to map (as a spatial polygon) the area that could have been jointly accessed by two individual animals in space ant time. The jPPA represents a spatial measure of spatial-temporal interaction.

Usage

```
jppa(traj1, traj2, t.int = 0.1 *
  as.numeric(names(sort(-table(ld(traj1)$dt)))[1]), tol = max(ld(traj1)$dt,
  na.rm = T), dissolve = TRUE, proj4string = CRS(as.character(NA)),
  ePoints = 360, ...)
```

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 `help(ltraj)`.
`traj2`	same as `traj1`.
`t.int`	(optional) time parameter (in seconds) used to determine the frequency of time slices used to delineate the joint activity space. Default is 1/10th of the mode of the temporal sampling interval from `traj1`. Smaller values for `t.int` will result in smoother output polygons.
`tol`	(optional) parameter used to filter out those segments where the time between fixes is overly large (often due to irregular sampling or missing fixes); which leads to an overestimation of the activity space via the PPA method. Default is the maximum sampling interval from `traj1`.
`dissolve`	logical parameter indicating whether (=`TRUE`; the default) or not (=`FALSE`) to return a spatially dissolved polygon of the joint activity space.
`ePoints`	number of vertices used to construct each PPA ellipse. More points will necessarily provide a more detailed ellipse shape, but will slow computation; default is 360.
`proj4string`	a string object containing the projection information to be passed included in the output `SpatialPolygonsDataFrame` object. For more information see the `CRS-class` in the packages `sp` and `rgdal`. Default is `NA`.
`...`	additional parameters to be passed to the function `dynvmax`. For example, should include options for `dynamic` and `method`; see the documentation for `dynvmax` for more detailed information on what to include here.

Details

The function `jppa` can be used to map areas of potential interaction between two animals. Specifically, this represents a measure of spatial overlap that also considers the temporal sequencing of telemetry points. In this respect it improves significantly over static measures of home range overlap, often used to measure static interaction, and can be considered as a spatial measure of dynamic interaction.

Value

This function returns a `SpatialPolygonsDataFrame` representing the joint accessibility space between the two animals.

See Also

dynvmax, dynppa

`ppaEllipse` *PPA Ellipse*

Description

Internal ellipse calculation function.

Usage

```
ppaEllipse(x, y, a, b, theta, steps)
```

Arguments

`x`	first coordinate
`y`	second coordinate
`a`	semi-major axis
`b`	semi-minor axis
`theta`	rotation angle of the ellipse (in radians)
`steps`	number of segments, from ePoints parameter in `dyn.ppa.hr`

Details

Internal function for calculating ellipses in time geographic analysis.

Value

This function returns a polygon ellipse.

See Also

dynppa

sma *Slow movement areas*

Description

The function `sma` computes the areas representing slow movement areas as described in the paper Nelson et al. (2014). Slow movement areas represent areas of sustained or intense habitat use, related to slow movement behaviours. Slow movement areas are defined by counting consecutive fixes within time geographic ellipses, and represented spatially as spatial polygons that are the union of included telemetry fixes within the slow movement area.

Usage

```
sma(traj, sma.keep = 1, sma.tol = 1, tol = max(ld(traj)$dt, na.rm = TRUE),
  proj4string = CRS(as.character(NA)), ePoints = 360, ...)
```

Arguments

`traj`	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 `help(ltraj)`.
`sma.keep`	an integer value indicating the number of slow movement areas to delineate, default is 1.
`sma.tol`	a value <= 1 indicating used when sma.keep > 1 to define how much overlap is allowed between SMA's, if sma.tol=1 no overlap is allowed, if sma.tol=0, any and all overlap is allowed. Typically something in between is most useful. Defaults to 1.
`tol`	(optional) parameter used to filter out those segments where the time between fixes is overly large (often due to missing fixes); which leads to an overestimation of the activity space via the PPA method. Default is the maximum sampling interval from `traj1`.
`proj4string`	a string object containing the projection information to be passed included in the output `SpatialPolygonsDataFrame` object. For more information see the `CRS-class` in the packages `sp` and `rgdal`. Default is `NA`.
`ePoints`	number of vertices used to construct each PPA ellipse. More points will necessarily provide a more detailed ellipse shape, but will slow computation; default is 360.
`...`	additional parameters to be passed to the function `dynvmax`. For example, should include options for `dynamic` and `method`; see the documentation for `dynvmax` for more detailed information on what to include here.

Details

The function `sma` can be used to map slow movement areas identifiable from wildlife telemetry data.of potential interaction between two animals. Slow movement areas can be ranked, according to their importance, which equates to consecutive time spent in an area. That is, the first slow movement area will be the area where the animal stayed the longest, and so on. Thus, slow movement areas can be useful for identifying where encamped behaviour or intensively exploited habitat on the landscape.

Value

This function returns a `SpatialPolygonsDataFrame` representing the joint accessibility space between the two animals.

See Also

dynvmax, dynppa

Index