

# wildlifeTG

October 5, 2014

**Type** Package  
**Title** Time Geographic Analysis of Wildlife Telemetry Data  
**Version** 0.1  
**Date** 2014-10-05  
**Author** Jed Long  
**Maintainer** Jed Long <jed.long@st-andrews.ac.uk>  
**Description** Description goes here.  
**License** GPL-3  
**Imports** sp,adehabitatLT,rgeos

## R topics documented:

wildlifeTG-package . . . . .	1
dyn.ppa.hr . . . . .	2
dyn.vmax . . . . .	3
j.dyn.ppa.hr . . . . .	5
jDynPPAHR . . . . .	6
ppaEllipse . . . . .	7
<b>Index</b>	<b>8</b>

---

wildlifeTG-package	<i>wildlifeTG - Time Geographic Analysis of Wildlife Telemetry Data</i>
--------------------	---

---

## Description

The Package wildlifeTG provides tools for performing dynamic time geographic analysis of animal tracking data. The functions provide useful tools for examining movement accessibility...

## Details

wildlifeTG's functions utilize the lttraj 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.

dyn.ppa.hr

*Dynamic PPA Home Range***Description**

The function `dyn.ppa.hr` computes the dynamic PPA home range of an animal. It can be used to delineate that area accessible to an animal given its set of telemetry locations. It incorporates dynamic behaviour into the calculation of the `vmax` parameter used to delineate the PPA home range.

**Usage**

```
dyn.ppa.hr(traj, tol = max(ld(traj)$dt, na.rm = TRUE), dissolve = TRUE,
           ePoints = 360, proj4string = NA, ...)
```

**Arguments**

<code>traj</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>tol</code>	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 <code>traj</code> .
<code>dissolve</code>	(logical) whether or not to dissolve output ellipse polygons to create a single output polygon, or keep the individual segment PPA ellipses. Default = <code>TRUE</code> .
<code>ePoints</code>	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.
<code>proj4string</code>	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> .
<code>...</code>	additional parameters to be passed to the function <code>dyn.vmax</code> . For example, should include method and/or window parameters, see the documentation for <code>dyn.vmax</code> for more detailed information on what to include here.

**Details**

The function `dyn.ppa.hr` represents an extension to an existing home range method – the PPA home range (Long and Nelson, 2012). The dynamic PPA home range improves upon this original method 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.

**Value**

This function returns a `SpatialPolygonsDataFrame` representing the dynamic PPA home range for an individual animal.

**References**

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

**See Also**

dyn.vmax, di.ppa.hr

---

dyn.vmax	<i>Dynamic calculation of the Vmax parameter</i>
----------	--

---

**Description**

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

**Usage**

```
dyn.vmax(traj, window = "focal", w = 9, class.col = "dt",
         method = "Robson", k = 5, alpha = 0.05, manualVmax = NA,
         vmaxtrunc = NA)
```

**Arguments**

traj	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> .
window	one of 'focal', 'cumulative', or 'class'; which signifies how to dynamically compute the <code>Vmax</code> parameter. See <b>Details</b> for more information on each of the choices.
w	(optional) window size (only used with <code>window = 'focal'</code> or <code>'cumulative'</code> ).
class.col	(optional) character indicating the name of the column in the <code>inflocs</code> dataframe of <code>traj</code> containing the categorized behavioural states of the animal (which can be stored as a character or numeric column).
method	method for computing the <code>Vmax</code> 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.

k	value for the $k$ parameter in the van der watt (1980) method; default is 5.
alpha	value for the $\alpha$ parameter if using upper or lower C.I. methods; default is 0.05.
manualVmax	Character name of column in traj storing user input column of vmax values (typically call the column dynVmax).
vmaxtrunc	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 `dyn.vmax` represents an intermediary function used to extend and improve upon an existing home range method – the PPA home range (Long and Nelson, 2012). Three options are available for computing the Vmax parameter dynamically. 1) `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.

2) `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.

3) `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 (see Patterson et al. 2008 for a review).

The use of the 'focal' or 'cumulative' 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, J.A., Nelson, T.A. 2012. Time geography and wildlife home range delineation. *Journal of Wildlife Management*. 76(2):407-413.
- Robson, D.S., Whitlock, J.H. 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

`dyn.ppa.hr`, `di.ppa.hr`

## Description

The function `jDynPPAHR` computes the joint activity space between two animals. It can be used to delineate that area jointly accessible to two individual animals in space and time.

## Usage

```
j.dyn.ppa.hr(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, ePoints = 360, ...)
```

## 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>t.int</code>	(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 <code>traj1</code> . Smaller values for <code>t.int</code> will result in smoother output polygons.
<code>tol</code>	(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 <code>traj1</code> .
<code>dissolve</code>	logical parameter indicating whether ( <code>=TRUE</code> ; the default) or not ( <code>=FALSE</code> ) to return a spatially dissolved polygon of the joint activity space.
<code>ePoints</code>	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.
<code>...</code>	additional parameters to be passed to the function <code>DynVmax</code> . For example, should include <code>window</code> and <code>method</code> ; see the documentation for <code>DynVmax</code> for more detailed information on what to include here.

## Details

The function `jDynPPAHR` 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 telemetry point. In this respect it improves significantly over static measures of home range overlap, often used to measure static interaction.

## Value

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

**See Also**

dyn.vmax, dyn.ppa.hr

---

jDynPPAHR

*Joint Activity Space (Dynamic PPA Home Range)*


---

**Description**

The function jDynPPAHR computes the joint activity space between two animals. It can be used to delineate that area jointly accessible to two individual animals in space and time.

**Usage**

```
jDynPPAHR(traj1, traj2, t.int = 0.1 *
  as.numeric(names(sort(-table(ld(traj1)$dt)))[1]), tol = max(ld(traj1)$dt,
  na.rm = T), vmaxtrunc = NA, dissolve = TRUE, 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 <code>help(ltraj)</code> .
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.
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.
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.
...	additional parameters to be passed to the function DynVmax. For example, should include method; see the documentation for DynVmax for more detailed information on what to include here.

**Details**

The function jDynPPAHR 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 telemetry point. In this respect it improves significantly over static measures of home range overlap, often used to measure static interaction.

**Value**

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

**See Also**

`dyn.vmax`, `dyn.ppa.hr`

---

`ppaEllipse`*PPA Ellipse*

---

**Description**

Internal ellipse calculation function.

**Usage**

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

**Arguments**

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

**Details**

Internal function for calculating ellipses in time geographic analysis.

**Value**

This function returns a polygon ellipse.

**See Also**

`dyn.ppa.hr`

# Index

`dyn.ppa.hr`, [2](#)

`dyn.vmax`, [3](#)

`j.dyn.ppa.hr`, [5](#)

`jDynPPAHR`, [6](#)

`ppaEllipse`, [7](#)

`wildlifeTG-package`, [1](#)