Package 'movementAnalysis'

November 10, 2012

Version 0.1

Date 2012/11/09

Depends R (>= 2.14.0), adehabitatLT, lmomco

Title Analysis of trajectory data using linear or Brownian motion model

Author Stef Sijben

Description Analysis of trajectory data. Contains functions for detecting movement patterns using different movement models, including the linear motion model and the Brownian bridge motion model (BBMM).

Maintainer Stef Sijben <s.m.a.sijben@student.tue.nl>

License GPL (>= 2)

R topics documented:

as.bbtraj *Working with Trajectories in 2D Space: the Class bbtraj*

Description

The class bbtraj stores trajectories of animals, similar to the class ltraj in package adehabitatLT. The difference is that a bbtraj object also stores the variances used when dealing with Brownian bridges. When converting a data set to an object of this class (using the function as.bbtraj) the most likely value for the diffusion coefficient is calculated and stored in the resulting object. bbFilterNA filters the missing measurements from a trajectory, removing bursts that are empty after this filtering.

Usage

```
as.bbtraj(xys, date = NULL, id, burst = id, typeII = TRUE, slsp = c("remove", "missing"))
bbFilterNA(tr)
```

Arguments

xys	a data.frame containing the x and y coordinates and variance of the relocations
date	for trajectories of type II, a vector of class POSIXct giving the date for each relocation. For trajectories of type I, this argument is not taken into account.
id	either a character string indicating the identity of the animal or a factor with length equal to nrow(xy)
burst	either a character string indicating the identity of the burst of relocations or a factor with length equal to nrow(xy)
typeII	logical. TRUE indicates a trajectory of type II (time recorded, e.g. radio-tracking), whereas FALSE indicates a trajectory of type I (time not recorded, e.g. sampling of tracks in the snow)
slsp	a character string used for the computation of the turning angles (see details)
tr	A trajectory of class ltraj or bbtraj

Value

ltraj is a list with one component per burst of relocations. Each component is a data frame with two attributes: the attribute "id" indicates the identity of the animal, and the attribute "burst" indicates the identity of the burst. Each data frame stores the following columns:

x	the x coordinate for each relocation
y	the y coordinate for each relocation
diff.coeff	The diffusion coefficient for the move. This has no meaning if the previous relocation is missing, but usually it is set to the same value for many relocations.
loc.var	The variance of the measured position
date	the date for each relocation (type II) or a vector of integer giving the order of the relocations in the trajectory.

dx	the increase of the move in the x direction. At least two successive relocations are needed to compute dx. Missing values are returned otherwise.
dy	the increase of the move in the y direction. At least two successive relocations are needed to compute dy. Missing values are returned otherwise.
dist	the length of each move. At least two successive relocations are needed to compute dist. Missing values are returned otherwise.
dt	the time interval between successive relocations
R2n	the squared net displacement between the current relocation and the first relocation of the trajectory
abs.angle	the angle between each move and the x axis. At least two successive relocations are needed to compute abs.angle. Missing values are returned otherwise.
rel.angle	the turning angles between successive moves. At least three successive relocations are needed to compute rel.angle. Missing values are returned otherwise.

References

Horne, J., Garton, E., Krone, S. and Lewis, J. Analyzing animal movements using Brownian bridges. *Ecology 88*, 9 (2007), 2354–2363.

See Also

ltraj, adehabitatLT

Examples

```
data("example_data", package="movementAnalysis")
example_data

tr <- as.bbtraj(data.frame(x=example_data$X, y=example_data$Y, var=example_data$StdDev^2),
date=example_data$DateTime, id=example_data$GroupID, burst=example_data$GroupDayNo)
tr

bbFilterNA(tr)
```

---

distance *Compute distance statistics*

Description

Density, distribution function, quantile function for the distances between entities at the given time(s), using the Brownian bridge movement model.

Usage

```
ddistance(d, tr, time)
pdistance(d, tr, time)
qdistance(p, tr, time)
```

Arguments

d	Vector of distances
p	Vector of probabilities
tr	The trajectory object
time	Vector of times

Value

ddistance computes the density, pdistance the distribution function and qnorm evaluates the quantile function for the requested parameters.

The functions return the requested values for each value of the first parameter (d or p) and for each of the requested times. The value is computed between each pair of IDs in the trajectory.

This means that the result is a 4 dimensional array, indexed by the IDs involved, the value of d or p and the time.

Examples

```
data("example_data", package="movementAnalysis")
tr <- as.bbtraj(data.frame(x=example_data$X, y=example_data$Y, var=example_data$StdDev^2),
date=example_data$DateTime, id=example_data$GroupID, burst=example_data$GroupDayNo)

# Compute the 5th and 95th percentile of the distance at two distinct times
qdistance(c(0.05, 0.95), tr, as.POSIXct(c("2011-01-18 15:15:15", "2011-01-19 16:30:00")))
```

---

encounterDistribution *Spatial distribution of encounters*

Description

Computes the expected duration of encounters at each location for every pair of IDs.

Usage

```
encounterDistribution(tr, threshold, xc, yc, timestepSize = 60)
```

Arguments

tr	The trajectory for which to compute the UD
threshold	The maximum distance at which an encounter occurs
xc	The x coordinates of the vertical grid lines
yc	The y coordinates of the horizontal grid lines
timestepSize	The difference between consecutive time steps, in seconds

Value

The return value is a list, indexed by two IDs. Each element of the list is a matrix representing the expected duration of encounters at each cell of the specified grid. The diagonal entries of the result list contain the utilization distribution of each ID, since an entity is always at a distance 0 from itself.

Warning

There seems to be some problem with the result being transposed, this needs further investigation. Until then, you can plot the transpose of the result using `image(t(ud[["BD","NH"]]))`.

Note

The `image` function has ugly colours, use the `col` attribute to define a better colour map.

Also note that this function may take a rather long time to complete, so please be patient, specify a sufficiently small grid or use a larger time step.

Examples

```
data("example_data", package="movementAnalysis")
tr <- as.bbtraj(data.frame(x=example_data$X, y=example_data$Y, var=example_data$StdDev^2),
date=example_data$DateTime, id=example_data$GroupID, burst=example_data$GroupDayNo)
tr <- bbFilterNA(tr) # Some operations in the following do not like NAs

# Define grid lines: equally spaced between the min and max coordinate in tr
xmin <- min(unlist(sapply(tr, function(b) { b$x })))
xmax <- max(unlist(sapply(tr, function(b) { b$x })))
xc <- seq(xmin, xmax, length.out=30)

ymin <- min(unlist(sapply(tr, function(b) { b$y })))
ymax <- max(unlist(sapply(tr, function(b) { b$y })))
yc <- seq(ymin, ymax, length.out=30)

# Compute the UD and plot the result for one ID
ud <- encounterDistribution(tr, 100, xc, yc)
image(ud[["BD","NH"]])
```

`encounterDuration` *Compute duration of encounters between groups*

Description

Computes the duration of encounters between each pair of groups over the whole measurement period of a trajectory. The user can select what movement model(s) to apply and whether the result should be on the level of bursts or IDs. `encounterDurationById` returns the duration of encounters for each pair of IDs, given the duration of encounters for each pair of bursts. This allows to obtain both types of result without recomputing everything.

Usage

```
encounterDuration(tr, threshold, model = c("BBMM", "linear"), byburst = FALSE, timestepSize = 60)
encounterDurationById(encounterDurationByBurst)
```

Arguments

tr	The trajectory to analyze
threshold	The maximum distance at which an encounter is detected
model	The movement models for which to compute the encounter duration
byburst	If TRUE, the result contains durations for pairs of bursts. If FALSE, the result contains encounter durations between pairs of IDs.
timestepSize	If a movement model requires numerical integration in the time dimension, this is the size of each time step.

Details

Since the duration of encounters is a random variable in the Brownian bridge movement model, this function cannot give exact results there. Instead, it reports the expected duration of encounters in the BBMM.

If you already have the encounter duration between bursts and you also want the encounter duration between IDs, you should use `encounterDurationById` instead of calling `encounterDuration` again with different parameters. They give identical results, but the former is much faster since it does not recompute all relevant durations.

Value

If `byburst == FALSE`, the result is a `data.frame` with one row for each interesting pair of bursts. A pair of bursts is interesting if they overlap in time, since otherwise the duration is always zero. The result contains the following fields:

id1,id2	The IDs of the bursts involved
burst1,burst2	The names of the bursts involved
	In addition there is one column for each model requested, named after the model. These columns contain the encounter duration according to that model.

If `byburst == TRUE`, the result is a 3 dimensional array, indexed by the two IDs and the movement model.

Examples

```
data("example_data", package="movementAnalysis")
tr <- as.bbtraj(data.frame(x=example_data$X, y=example_data$Y, var=example_data$StdDev^2),
date=example_data$DateTime, id=example_data$GroupID, burst=example_data$GroupDayNo)

d <- encounterDuration(tr, 100, byburst=TRUE)
d

#   id1 id2  burst1  burst2      BBMM    linear
```

```
# 1  NH  BD NH_1416 BD_1416  4507.143  5420.426
# 2  BD  NH BD_1417 NH_1417 11693.703 15058.015

encounterDurationById(d)

# , , BBMM
#
#          NH       BD
# NH       NA 16200.85
# BD 16200.85       NA

# , , linear
#
#          NH       BD
# NH       NA 20478.44
# BD 20478.44       NA

encounterDuration(tr, 100)
# Same result, but needs to recompute everything
```

`position` *Evaluate position parameters at the given times*

Description

Given a trajectory of class `bbtraj` and a list of date/times (as timestamps or objects of class `POSIXct`), this function computes the distibution parameters of the position for each group in the trajectory at each of the requested times. The distribution parameters are the mean location and the variance of the location.

Usage

```
position(tr, time)
```

Arguments

`tr`	An object of class `bbtraj`
`time`	A list of date/times at which to evaluate the position parameters.

Details

If a requested time does not have a relocation, the values are interpolated in the usual way for the Brownian bridge movement model.

Value

An array indexed by ID, parameter name, time.

See Also

bbtraj

Examples

```
data("example_data", package="movementAnalysis")
tr <- as.bbtraj(data.frame(x=example_data$X, y=example_data$Y, var=example_data$StdDev^2),
date=example_data$DateTime, id=example_data$GroupID, burst=example_data$GroupDayNo)

position(tr, as.POSIXct(c("2011-01-19 08:30:00")))

# , , 1295422200
#
#           x       y      var
# BD 322487.6 6900484 1133.653
# NH 323012.9 6900720 1192.534
```

---

utilizationDistribution

Compute utilization distribution for a trajectory

Description

This function computes the utilization distribution (UD) for each of the IDs present in the trajectory. The user specifies the grid on which the UD is evaluated and may also specify the size of the time step in the numerical integration.

Usage

```
utilizationDistribution(tr, xc, yc, timestepSize = 60)
```

Arguments

tr	The trajectory for which to compute the UD
xc	The x coordinates of the vertical grid lines
yc	The y coordinates of the horizontal grid lines
timestepSize	The difference between consecutive time steps, in seconds

Value

Returns a list, indexed by the IDs in tr. Each element of the list is a matrix, indexed by the coordinates specified in xc and yc.

Warning

There seems to be some problem with the result being transposed, this needs further investigation. Until then, you can plot the transpose of the result using image(t(ud[["BD","NH"]])).

Note

The image function has ugly colours, use the col attribute to define a better colour map.

Examples

```
data("example_data", package="movementAnalysis")
tr <- as.bbtraj(data.frame(x=example_data$X, y=example_data$Y, var=example_data$StdDev^2),
date=example_data$DateTime, id=example_data$GroupID, burst=example_data$GroupDayNo)
tr <- bbFilterNA(tr) # Some operations in the following do not like NAs

# Define grid lines: equally spaced between the min and max coordinate in tr
xmin <- min(unlist(sapply(tr, function(b) { b$x })))
xmax <- max(unlist(sapply(tr, function(b) { b$x })))
xc <- seq(xmin, xmax, length.out=100)

ymin <- min(unlist(sapply(tr, function(b) { b$y })))
ymax <- max(unlist(sapply(tr, function(b) { b$y })))
yc <- seq(ymin, ymax, length.out=100)

# Compute the UD and plot the result for one ID
ud <- utilizationDistribution(tr, xc, yc)
image(ud[["BD"]])
```

Index