Package ‘move’

July 4, 2012

Type Package

Title Visualizing and analyzing animal track data

Version 0.5.2

Date 2012-05-07

Author Bart Kranstauber <bart.kranstauber@uni-konstanz.de>, Kamran Safi <ksafi@orn.mpg.de>, Marco Smolla <msmolla@orn.mpg.de>

Maintainer Marco Smolla <msmolla@orn.mpg.de>

Description move is a package that contains functions to access movement data stored in movebank.org as well as tools to visualize and statistically analyse animal movement data. Move addresses movement ecological questions.

License GPL (>=3)

URL http://move.r-forge.r-project.org/

LazyLoad yes

Depends sp, raster, rgdal, methods, geosphere, R (>= 2.14.0)

Suggests RgoogleMaps, RCurl

Collate ClassMove.R ClassMoveStack.R ClassDBMvar.R ClassDBBMM.R WebImport.R emd.R main.R

Archs i386, x86_64

R topics documented:

`move-package` *An overview of the functions in this package*

Description

move is a package that contains functions to access movement data stored at `www.movebank.org` as well as tools to visualize and statistically analyse animal movement data. Move addresses movement ecological questions.

Details

The package implements classes for movement data and supports

- Creation of Move objects (see Move-class) representing animals and their track
- Calculation of utilization densities using the dynamic Brownian bridge Movement Model
- Plotting tracks, utilization densities and contours
- Access to raster, n.col, projection and coordinates
- Different CRS projection methods such as longlat or aeqd

I. Creating Move objects

Move objects can be created from files with the function:

`move`	To create a Move object

II. Calculation of the utilization density

With the function below the dynamic Brownian Bridge Movement Model calculates the utilization density from a Move object:

`brownian.bridge.dyn`	To calculate the utilization density

III. Accessing values

`coordinates`	Track-coordinates of the Move Object
`as.data.frame`	A data.frame with the important data of the Move Object
`n.locs`	The number of locations
`time.lag`	The time lags between the locations
`projection`	The projection method of the track/raster

IV. Plotting data

The track or the utilization density can be plotted with the following functions:

`plot`	plots the utilization density with fixed width and height ratio (see DBBMM-class) or the track (see Move-class)
`image`	plots the utilization density fitted to the window
`contour`	adds the contours of utilization densities to a plot

Author(s)

Bart Kranstauber, Marco Smolla, Kamran Safi

Maintainer: Bart Kranstauber <`kranstauber@orn.mpg.de`>, Marco Smolla <`msmolla@orn.mpg.de`>

References

CRAN: `http://move.r-forge.r-project.org/`

`as.data.frame` *Return a Data Frame*

Description

Function to create a data.frame with the spatial data frame information from a Move object.

Usage

```
  ## S4 method for signature 'Move'
as.data.frame(x,...)
```

Arguments

x an object of the Move-class

... additional arguments to be passed to or from methods

Details

as.data.frame extracts the sdf argument from a Move object (see Move-class)

Author(s)

Marco Smolla

Examples

```
## create a move object
data <- move(system.file("extdata","leroy.csv",package="move"),proj=CRS("+proj=longlat"))

## returns a data.frame with all information stored in the spatial data frame of the move object
df <- as.data.frame(data)
```

---

brownian.bridge.dyn *Creates a DBBMM object*

Description

The brownian.bridge.dyn function uses a Move object (see Move-class) to calculate the utilization density (UD) of the given track. It uses the dynamic Brownian Bridge Movement Model (dBBMM) to do so. The dBBMM has the advantage over the other Brownian Bridge Movement Model that changes in behaviour are accounted for. It does so by using the behavioral change point analysis in a sliding window. For details see Kranstauber et al. (2012).

Usage

```
  brownian.bridge.dyn(object, raster, dimSize, location.error,
                 margin=11, time.step=NULL, window.size=31, ext, ...)
```

Arguments

object a Move-class object, which can be created with the function move

raster can be a RasterLayer object or numeric. If it is a RasterObject and dimSize is not set, brownian.bridge.dyn starts to calculate the UD. If raster is numeric, it describes the cell.size of the raster. Default is 1

dimSize can be numeric. If there is no input for raster, dimSize describes the number of cells of the largest x or y dimenstion. Default is 10

location.error required numeric value. Describes the error of the location (sender/receiver) system in map units.

`margin` The margin used for the behavioral change point analysis.

`time.step` The size of the time steps taken for the calculation of the utilization density in minutes, if left null 15 steps are taken in the shortest time interval.

`window.size` The size of the moving window along the track. Larger windows provide more stable/accurate estimates of the brownian motion variance but are less well able to capture more frequent changes in behaviour.

`ext` numeric value >0; extension factor. Decribes the amount of extension of the bounding box of the animal track. It can be one (same extension into all four directions), two (first x, then y directional extension) or four numbers (xmin, xmax, ymin, ymax extension). Default is .25 (extends the bounding box by 25%)

`...` for additional arguments

Details

There are four ways to launch the brownian.bridge.dyn function which are as follows:

1. Serve a raster
A RasterLayer object is set for the raster argument which is then used to calculate the UD.

2. Set the cell size
To set the cell size set a numeric value for the raster argument but skip the dimSize. The numeric raster argument is used as the cell sizes of the raster. A RasterLayer object is created and returned to the same function.

3. Set the number of cells (col/row)
To set the number of cells along the largest dimension a numeric dimSize argument is set instead of the raster argument. A RasterLayer object is created and returned to the same function.

4. Using default raster
When there are no values set, the default raster value is used to calculate and create a RasterLayer object, which is returned to the same function. Note: depending on the size of the area of interest, the default cell size value can result in a large number of cells which may take a very long time to calculate!

The function prints an estimate of the size of the computational task ahead. This can give an indication of how long the computation is going to take. It should scale roughtly linearly with the duration of the computations. In our experience $1*10^9$ takes about a minute with a normal laptop.

Note

Note that the first few and last few segments of the trajectory are omitted in the calculation of the UD since a lower number of estimates for the Brownian motion variance are obtained for those segments.

Author(s)

Marco Smolla

References

Kranstauber, B., Kays, R., LaPoint, S. D., Wikelski, M. and Safi, K. (2012), A dynamic Brownian bridge movement model to estimate utilization distributions for heterogeneous animal movement. Journal of Animal Ecology. doi: 10.1111/j.1365-2656.2012.01955.x

Examples

```
## create a move object
data <- move(system.file("extdata","leroy.csv",package="move"),proj=CRS("+proj=longlat"))

## change projection method to aequd and center the coordinate system to the track
data2 <- spTransform(data, CRSobj="+proj=aeqd", center=TRUE)

## create a DBBMM object
#dbbmm <- brownian.bridge.dyn(object=data2, location.error=23.5, dimSize=45, ext=.3, time.step=600)
```

`burstTrack` *Bursting a track*

Description

Bursting a track by specified variable

Usage

```
  ## S4 method for signature 'Move,numeric'
burstTrack(object, by, breaks=5, sizeFUN="relTime")
```

Arguments

`object` a Move object

`by` column indicator like `data$behavior` that is used to burst the track

`breaks` defines the number of steps by which the relative values, e.g. relative time, is divided. Default value is 5.

`sizeFUN` specify a function to calculate the sizes of the centroids like `function(ll) lapply(lapply(ll, le` DOES NOT WORK AT THE MOMENT! Default is relative time of the burst segemnt compared to the whole track

Details

The burstTrack function bursts (divides) a track in segments that are specified by the by attribute. If a colum with behavioral classfication is annotated to the coordinates of the Move object, the track can be bursted by that column. For every segment an indicator (a circle) is set in the middle of the segment. The size of this indicator depends on the sizeFUN and is by default defined by the relative time compared to the whole duration.

Note

sizeFUN does not work at the moment

Author(s)

Marco Smolla

Examples

```
##here we need a annotated csv file
```

---

contour *Contour plot*

Description

Contour plot of a RasterLayer from a DBBMM object.

Usage

```
  ## S4 method for signature '.UD'
contour(x, ...)
```

Arguments

x	an object of the DBBMM-class
...	additional arguments like levels and nlevels, see details

Details

The contour function creates a shape of the area in which there is a 90% probability that the animal will be located. One or several probabilities can be set with `levels`. Choose values between 0 and 1. At least one level must be added, there is no default value set. You can also use `nlevel` to set a number of fixed distance levels.
To change parameters of the contour or line plotting use the usual parameters of the `plot` function.

Author(s)

Marco Smolla

Examples

```
## Not run:
## Create a DBBMM object to use this function
## See the examples of the brownian.bridge.dyn function
## Here we just load a previous session, where we already created a DBBMM object
load(system.file("extdata", "leroy.RData", package="move"), .GlobalEnv)

## to add a 50% and 90% contour to a plot from DBBMM object dbbmm
plot(dbbmm)
contour(dbbmm, levels=c(.5,.9), add=TRUE, plot=TRUE)

##to store the contours as a SpatialLinesDataFrame use
cnt  <- contour(dbbmm, levels=c(.4,.9))

##print the contour on a google map +add the track
```

```
require("RgoogleMaps")
cnt  <- contour(x=dbbmm, y=data, levels=c(.4,.9), google=TRUE)
cnt  <- contour(x=dbbmm, y=data, levels=c(.4,.9), google=TRUE, track=TRUE)

## End(Not run)
```

coordinates *Extract the track coordinates from a Move object*

Description

The coordinates method extracts the coordinates of the track within a Move object.

Usage

```
  ## S4 method for signature 'Move'
coordinates(obj,...)
```

Arguments

`obj`	A valid Move object
`...`	Additional arguments, see Details

Details

`obj` The move object includes a special data frame with the track coordinates and a time stamp. They are returned as a matrix.

Author(s)

Marco Smolla

Examples

```
## create a move object
data <- move(system.file("extdata","leroy.csv",package="move"), proj=CRS("+proj=longlat"))

## extract the coordinates
coords <- coordinates(data)
```

`DBBMM-class` *The DBBMM class*

Description

The DBBMM object is created within the brownian.bridge.dyn function from a Move object. It includes beneath others a raster object and probabilities.

Slots

DBMvar Object of class `"dBMvarianceTmp"`: includes the window.size, margin, means, in.windows, break.list, and points of interest

crs part of the Raster-class

ext the extension factor set by the user

data part of the Raster-class

extent part of the Raster-class

file part of the Raster-class

history part of the Raster-class

layernames part of the Raster-class

legend part of the Raster-class

method stores the method that was used to calculate the utilization distribution (UD), e.g. dynamic Brwonian Bridge

ncols part of the Raster-class

nrows part of the Raster-class

rotated part of the Raster-class

rotation part of the Raster-class

title part of the Raster-class

z part of the Raster-class

Methods

`contour` `signature(object = "DBBMM")`: adds a contour line to a plot

`image` `signature(object = "DBBMM")`: plots the raster from a DBBMM object with fixed cell size ratio

`plot` `signature(object = "DBBMM")`: plots the raster from a DBBMM object with resize insenstive proportions

`proj4string` `signature(object = "DBBMM")`: extracts the projection method of the raster stored within the DBBMM object

`raster` `signature(object = "DBBMM")`: extracts the raster from the DBBMM object

`outerProbability` `signature(object = "DBBMM")`: calculates the animal occurence probabilities at the border of the raster

Author(s)

Marco Smolla

`DBBMMStack-class` *The DBBMMStack class*

Description

The DBBMMStack object is created within the brownian.bridge.dyn function from a Move object. It includes beneath others a raster object and probabilities.

Slots

DBMvar Object of class `"dBMvariance"`: includes the break.list and points of interest

crs part of the Raster-class

ext the extension factor set by the user

extent part of the Raster-class

filename part of the Raster-class

layers part of the Raster-class

layernames part of the Raster-class

method the method that was used to calculate the utilization distribution, e.g. dynamic Brwonian Bridge

ncols part of the Raster-class

nrows part of the Raster-class

rotated part of the Raster-class

rotation part of the Raster-class

title part of the Raster-class

z part of the Raster-class

Methods

`contour` `signature(object = "DBBMMStack")`: adds a contour line to a plot

`image` `signature(object = "DBBMMStack")`: plots the raster from a DBBMMStack object with fixed cell size ratio

`plot` `signature(object = "DBBMMStack")`: plots the raster from a DBBMMStack object with resize insenstive proportions

`proj4string` `signature(object = "DBBMMStack")`: extracts the projection method of the raster stored within the DBBMMStack object

`raster` `signature(object = "DBBMMStack")`: extracts the raster from the DBBMMStack object

`outerProbability` `signature(object = "DBBMMStack")`: calculates the animal occurence probabilities at the border of the raster

Author(s)

Marco Smolla

emd *Earth Movers Distance between DBBMM objects*

Description

Calculates the Earth Movers Distance between two rasters stored in DBBMM objects

Usage

```
  ## S4 method for signature ’RasterLayer,RasterLayer,numeric,logical,logical’
emd(x, y, threshold, integer, greatcircle)
  ## S4 method for signature ’DBBMM,DBBMM,numeric,logical,logical’
emd(x, y, threshold, integer, greatcircle)
```

Arguments

`x`	a DBBMM object
`y`	a DBBMM object
`threshold`	...
`integer`	...
`greatcircle`	...

Details

...

Note

...

Author(s)

Marco Smolla

Examples

```
  ## Not run:
    load(system.file(”extdata”, ”leroy.RData”, package=”move”), .GlobalEnv)

## End(Not run)
```

getMovebankAnimals *Animals, tags and IDs in a Movebank study*

Description

Returns the animals, their tags and IDs from a Movebank study

Usage

```
getMovebankAnimals(study, login)
```

Arguments

`study`	a character string (study name) or the numeric study ID as it is stored on Movebank
`login`	an object of the `MovebankLogin-class`, if empty you'll be asked to enter your username or password

Details

The function returns a `data.frame` that incldues the `animalID, animalName, id, sensor_type_id` and `tag_id` from the requested study.

Note

See the 'browesMovebank' vignette (move website download section) for more information about security and how to use Movebank from within R.

Author(s)

Marco Smolla

getMovebankData *Download data from Movebank*

Description

getMovebankData downloads the location and timestemp columns of an animal of study stored in Movebank and creates a Move object

Usage

```
getMovebankData(study, animalName, login, moveObject = TRUE, ...)
```

Arguments

`study`	character, full name of the study, as stored on Movebank
`animalName`	chracter, name of the individual as stored on Movebank
`login`	a `MovebankLogin`, if empty you'll be asked to enter your username or password
`moveObject`	logical, if FALSE a data.frame with coordiantes and timestamps is returned. Default is TRUE
`...`	passing on additional arguments

Details

With the getMovebankData function you can access the location and timestamp data of a particular animal within a certain study. By default `moveObject` is `TRUE`. Thus the function returns a Move object if an `animalName` is given. If there is no `animalName`, the whole study is downloaded and stored as a MoveStack object. Remember that you need an account at Movebank.org, see `movebankLogin`.

Note

See the 'browesMovebank' vignette (move website download section) for more information about security and how to use Movebank from within R.

Author(s)

Marco Smolla

`getMovebankID` *Study ID*

Description

Returns the study ID of a certain study stored on Movebank

Usage

```
getMovebankID(x,login)
```

Arguments

`x`	character, full name of the study, as stored on Movebank
`login`	an object of the `MovebankLogin-class`, if empty you'll be asked to enter your username or password

Details

The getMovebankID function returns the ID of a study as it is stored on Movebank.org.

Note

See the 'browesMovebank' vignette (move website download section) for more information about security and how to use Movebank from within R.

Author(s)

Marco Smolla

`getMovebankSensors` *Information about Movebank sensors*

Description

getMovebankSensors returns information about sensors in a study.

Usage

```
getMovebankSensors(study, login)
```

Arguments

`study`	a character string (study name) or the numeric study ID as it is stored on Movebank, if missing the function returns a data.frame with information about all sensor types in Movebank
`login`	an object of the `MovebankLogin-class`, if empty you'll be asked to enter your username or password

Details

The function returns either information about all sensor types that are available on Movebank or the sensor IDs corresponding to to the animal IDs in a specific study.

Note

See the 'browesMovebank' vignette (move website download section) for more information about security and how to use Movebank from within R.

Author(s)

Marco Smolla

`getMovebankSensorsAttributes` *Available sensor attributes*

Description

This function returns all attributes of the sensors of the requested study.

Usage

```
getMovebankSensorsAttributes(study, login)
```

Arguments

`study`	a character string (study name) or the numeric study ID as it is stored on Movebank
`login`	an object of the `MovebankLogin-class`, if empty you'll be asked to enter your username or password

Details

Returns the attributes of the sensors of a study, i.e. which sensor has which id and which data stored (e.g. GPS sensors store longitute and latitude locations, and timestamps and have 673 as their ID on Movebank).

Note

See the 'browesMovebank' vignette (move website download section) for more information about security and how to use Movebank from within R.

Author(s)

Marco Smolla

`getMovebankStudies` *All studies on Movebank*

Description

Returns all studies available on Movebank

Usage

```
getMovebankStudies(login)
```

Arguments

`login`	an object of the `MovebankLogin-class`, if empty you'll be asked to enter your username or password

Details

Returns all studies available on Movebank.

Note

See the 'browesMovebank' vignette (move website download section) for more information about security and how to use Movebank from within R.

Author(s)

Marco Smolla

`getMovebankStudy` *Returns study information*

Description

This function returns information about the requested study like the authors of that study, licence type, citation and more.

Usage

```
getMovebankStudy(study, login)
```

Arguments

`study`	a character string (study name) or the numeric study ID as it is stored on Movebank
`login`	an object of the `MovebankLogin-class`, if empty you'll be asked to enter your username or password

Note

See the 'browesMovebank' vignette (move website download section) for more information about security and how to use Movebank from within R.

Author(s)

Marco Smolla

`leroy` *GPS track data from a fisher*

Description

This file includes spatial data from a fisher (*Martes pennanti*). It can be used to test the different functions from the move package.

These location data were collected via a 105g GPS tracking collar (manufactured by E-obs gmbh) and programmed to record the animal's location every 15 minutes, continuously. The collar was deployed from 10 February 2009 through 04 March 2009 on an adult, resident, male fisher, in New York, USA. Scott LaPoint tracked this fisher for part of his dissertation research hoping to better understand how fishers were moving throughout this developed landscape.

Author(s)

Marco Smolla

References

For more information, contact Scott LaPoint <`sdlapoint@gmail.com`>

Examples

```
## create a Move object from the data set
data <- move(system.file("extdata","leroy.csv",package="move"),proj=CRS("+proj=longlat"))
```

---

lines *Plotting a track as lines*

Description

Function for plotting a recorded track from a Move object as lines

Usage

```
  ## S4 method for signature 'Move'
lines(x,add=FALSE,...)
```

Arguments

x	Move object
add	if TRUE the lines are added to a plot
...	arguments to be passed on, e.g. col for color, or add to add the lines to a plot.

Author(s)

Marco Smolla

Examples

```
  load(system.file("extdata", "leroy.RData", package="move"), .GlobalEnv)

  #plot the track of a single individual, with additional parameters
  lines(data, col="#428536", lwd=2)
```

---

move *Create a Move object*

Description

The move method creates a Move object from a Movebank file. Additionally the projection method must be set with the 'proj' argument.
An alternative to this method is to import one's own data, and specify the location and timestamp columns.

Usage

```
## S4 method for signature 'character,ANY,ANY,ANY,ANY,ANY'
move(x, proj)
## S4 method for signature 'numeric,numeric,POSIXct,data.frame,CRS,character'
move(x, y, time, data, proj, animal, species="sp", study="std",...)
```

Arguments

`x` Full path to the file location, OR column indicator for non-Movebank data for the x coordinates; e.g. `data$x`

`y` column indicator for non-Movebank data for the y coordinates

`time` column indicator for non-Movebank data for the time stamps, with POSIXct conversion, i.e. `as.POSIXct(data$timestamp, format="%Y-%m-%d %H:%M:%S", tz="UTC")`

`data` full parh of the location of the non-Movebank file

`proj` projection method for non-Movebank data; requires a valid CRS (see `CRS-class`) object, like CRS("+proj=longlat +ellps=WGS84")

`animal` animal ID or name, either single character or a vector with length of the number of coordinates

`species` name of the species

`study` name of the study

`...` Additional arguments

Details

There are two ways to import data to make a Move object. Either the path to a file downloaded from `www.Movebank.org` is set for the x argument, or the file is first imported with the base `read.csv` function and then the columns with the x and y coordinates, and the timestamp, as well as the whole data.frame of the imported data are given to the `move` function.

Note

The imported data set is checked whether it is in a Movebank format. If this is not the case, you have to use the alternative import for non-Movebank data.
Because the SpatialPointsDataFrame function that creates the spatial data frame (`sdf`) of the `Move` object can not process NA location values, all rows with NA locations are omitted. A list of the omitted timestamps is stored in the `timesMissedFixes` argument of the `Move` object.
If the data include double timestamps check your data for valitdity. You may want to consider a function to delete double timetamps, like:
`data <- data[which(!duplicated(data$timestamp)), ]`

Author(s)

Marco Smolla

Examples

```
## create a move object from a Movebank csv file
data <- move(system.file("extdata","leroy.csv",package="move"),proj=CRS("+proj=longlat"))

## create a move object from non-Movebank data
file <- read.table(system.file("extdata","leroy.csv",package="move"), header=TRUE, sep=",", dec=".")
data <- move(x=file$location.long, y=file$location.lat, time=as.POSIXct(file$timestamp, format="%Y-%m-%d %H
```

Move-class *The Move class*

Description

The Move object is created with the `move` function. It is the required object class for the `brownian.bridge.dyn` function which calculates the utilization density.

Slots

animal Object of class `"character"`: animal ID or name;

species Object of class `"character"`: species name;

dateCreation Object of class `"numeric"`: time stamp when the file was downloaded;

study Object of class `"character"`: name of the study;

citation Object of class `"character"`: how to cite the study;

license Object of class `"character"`: the license under which the data were publicated;

timesMissedFixes Object of class `"POSIXct"`: stores the timestamps of lines of the data set that were removed because they included NA locations

bbox belongs to the SpatialPointsDataFrame

coords coordinates of the track, belongs to the SpatialPointsDataFrame

coords.nrs belongs to the SpatialPointsDataFrame

data additional data of that object that is stored in the SpatialPointsDataFrame

proj4string projection of the coordinates

timestamps timestamps according to the coordinates

Methods

`move` `signature(object = "Move")`: creates a Move object

`getMovebankData` `signature(object = "character")`: creates a Move object by accessing Movebank

`spTransform` `signature(object = "Move")`: transforms coordinates to a different projection method

`show` `signature(object = "Move")`: prints a summary of all data stored in the Move object

`as.data.frame` `signature(object = "Move")`: extracts the spatial data frame

`coordinates` `signature(object = "Move")`: extracts the coordinates only from the Move object

`proj4string` `signature(object = "Move")`: extracts the projection method from the Move object

`time.lag` `signature(object = "Move")`: calculates time lags between coordinates

`n.locs` `signature(object = "Move")`: calculates number of locations

`plot` `signature(object = "Move")`: plots the track of the animal

Author(s)

Marco Smolla

movebankLogin *Login into Movebank*

Description

Creates an object that can be used with all Movebank browsing functions.

Usage

```
## S4 method for signature 'character,character'
movebankLogin(username,password)
```

Arguments

`username`	Your Movebank username
`password`	Your Movebank password

Details

Use this function to login to Movebank. After you logged in, you can use the Movebank browsing functions from the move package.

Note

If you do not have the RCurl package installed `movebankLogin` will store your username and password in the object you assign it to. Furthermore, if you have no RCurl a http connection is used to retrieve data from Movebank instead of the more secure https protocol.

Author(s)

Marco Smolla

Examples

```
## Not run:
##first create the login object
login <- movebankLogin{username="xxx", password="zzz"}

##and than use it with Movebank browsing functions
getMovebankStudies(login)

## End(Not run)
```

MovebankLogin-class *The MovebankLogin class*

Description

The MovebankLogin object is needed for every Movebank browsing function. Alternatively, one can also chose to enter the username and password everytime one uses one of the browsing functions.

Slots

username username as character

password password as character

rcurl logical value, whether RCurl is installed or not

Methods

movebankLogin `signature(object = "character")`: creates a MovebankLogin object

Author(s)

Marco Smolla

moveStack *Creating a MoveStack*

Description

Either stacks a list of Move objects or creates a MoveStack from a Movebank csv file

Usage

```
  ## S4 method for signature 'list'
moveStack(x)
  ## S4 method for signature 'character'
moveStack(x, proj)
```

Arguments

`x`	a list of Move objects or the path to a csv file as character
`proj`	the projection of the coordinates

Details

This function stacks single Move objects to a MoveStack object, or creates a MoveStack from a Movebank file with several individuals.

Author(s)

Marco Smolla

Examples

```
  ## Not run:
    we need more than one test file

## End(Not run)
```

`MoveStack-class` *The MoveStack class*

Description

The MoveStack object is created within the brownian.bridge.dyn function from a Move object. It includes beneath others a raster object and probabilities.

Slots

citation Object of class `"dBMvariance"`: includes the break.list and points of interest, ...;

bbox belongs to the SpatialPointsDataFrame

coords the extension factor set by the user

coords.nrs belongs to the SpatialPointsDataFrame

data additional data of that object that is stored in the SpatialPointsDataFrame

dateCreation date and timestamp when this object was created

idData additional data to all individuals

license the license terms of the used track material

proj4string projection of all coordinates

study name of the study

timestamps timestamps according to the coordinates

trackId a vector that indicates, which data, coordinates and timestamps belong to an individual

Methods

Methods defined with class "MoveStack" in the signature:

`[` `signature(x="MoveStack")`: select subset ...

`getMovebankData` `signature(object = "character")`: creates a MoveStack object by accessing Movebank

Author(s)

Marco Smolla

n.locs *Extract the number of locations of a track from a Move object*

Description

The n.locs method returns the number of locations of a track from a Move object.

Usage

```
  ## S4 method for signature 'Move'
n.locs(obj)
```

Arguments

obj	A Move object

Details

`obj` a Move object

Author(s)

Marco Smolla

outerProbability *Calculates the probabilities at the edges of a raster*

Description

The outerProbability method calculates the summed probabilitiy of the cells at the border of a raster

Usage

```
  ## S4 method for signature 'RasterLayer'
outerProbability(raster,border)
```

Arguments

raster	a RasterLayer object that has values for the raster cells
border	numeric from 0 to 1; ratio of the number of columns at the border relative to the whole raster from which the probabilities should be summed up

Author(s)

Marco Smolla

Examples

```
load(system.file("extdata", "leroy.RData", package="move"), .GlobalEnv)

## calculate the probabilities of 20% of the raster at
## the border from the DBBMM object dbbmm
outerProbability(dbbmm, border=.2)
"[1] 0.001669491"
```

---

plot *Plotting a track or a raster*

Description

Function for plotting a recorded track from a Move object or the probability values from a DBBMM object

Usage

```
  ## S4 method for signature 'Move,missing'
plot(x, google = FALSE, maptype = "terrain", ...)
  ## S4 method for signature 'MoveStack,missing'
plot(x, google = FALSE, maptype = "terrain", ...)
```

Arguments

`x`	Move or a DBBMM object
`google`	logical argument for a Move object. If TRUE, a map from googlemaps is downloaded and the track is plotted on top
`maptype`	defines the type of map to construct. There are several possible maptype values, including satellite, terrain, hybrid, and mobile.
`...`	arguments to be passed to methods, such as graphical parameters, and the logical add argument (see `par`)

Details

If `x` is a Move object a track is plotted with points and lines. The track can be added to another plot with the `add = TRUE`.
If `x` is a DBBMM object its raster object is plotted with the corresponding cell values. Unlike the `image` function, the cell size ratio keeps the same when the plot window is resized.
The `google` argument works together with a Move object. If `TRUE` the GetMap function from the RgoogleMaps package generates a URL to download a map that fits with the track. After downloading the map, the track is plotted on top of the map.

Note

If you want to combine e.g. a track and a raster, or a track and a google map, make sure that both objects have the same projection method. Use `proj4string` to find it out and `spTransform` to change the projection method of a Move object.

Author(s)

Marco Smolla

Examples

```
load(system.file("extdata", "leroy.RData", package="move"), .GlobalEnv)

## plot a Move object
plot(data)

#to plot a google map with overlayed track
plot(data, google=TRUE)

## add a track from a Move object to a plot
lines(data, add=TRUE)

## plot the raster of a DBBMM object
plot(dbbmm)
```

---

plotBursts *Plotting the centroids of a track*

Description

Plotting centroids ...

Usage

```
  ## S4 method for signature 'list'
plotBursts(object, add=TRUE, ...)
  ## S4 method for signature 'SpatialPointsDataFrame'
plotBursts(object, add, ...)
```

Arguments

`object`	a SpatialPointsDataFrame or a list of these that include coordinates and color, and size of the centroid indicators
`add`	logical, if FALSE a new plot is generated. Default value is TRUE
`...`	additional plot attributes

Details

The SpatialPointsDataFrame that should be used here is generated by `burstTrack`.

Note

...

Author(s)

Marco Smolla

Examples

```
  ## Not run:
we need an anotated file here

## End(Not run)
```

`points` *Plotting a track as points*

Description

Function for plotting a recorded track from a Move object as points

Usage

```
  ## S4 method for signature 'Move'
points(x,add=FALSE,...)
```

Arguments

`x`	Move or a DBBMM object
`add`	if TRUE the points are added to a plot
`...`	arguments to be passed on, e.g. `col` for color, or `add` to add the points to a plot.

Author(s)

Marco Smolla

`raster` *Extract raster from DBBMM*

Description

raster plot of a RasterLayer from a DBBMM object.

Usage

```
  ## S4 method for signature 'DBBMM'
raster(x)
```

Arguments

`x`	a DBBMM object

Details

The raster function extraxcts the raster object from the DBBMM object.

Value

An object from class RasterLayer is returned.

Author(s)

Marco Smolla

raster2contour *Convert raster to contour lines*

Description

The funciton converts a raster object to a SpatialLinesDataFrame.

Usage

```
  ## S4 method for signature 'RasterLayer'
raster2contour(x, ...)
```

Arguments

x	a RasterLayer or a DBBMM object
...	additional arguments that are passed on from other functions

Details

The raster2contour function creates a SpatialLinesDataFrame from a given raster or DBBMM object. This allows to reproject the contours to different projections.

Author(s)

Marco Smolla

Examples

```
load(system.file("extdata", "leroy.RData", package="move"), .GlobalEnv)

raster2contour(dbbmm)
```

`searchMovebankStudies` *Search for a study*

Description

Searches for a study within movebank

Usage

```
searchMovebankStudies(x, login, sensor = FALSE)
```

Arguments

`x`	a character string to search within the Movebank study names
`login`	an object of the `MovebankLogin-class`, if empty you'll be asked to enter your username or password
`sensor`	logical, if TRUE you can search for studies using a certain sensor type. Default is FALSE (not implemented yet)

Details

The search function searches explicitly for the entered phrase. If you for example type 'Goose' it will not show you studies including 'goose'. So rather search for 'oose' to find both.

Note

See the 'browesMovebank' vignette (move website download section) for more information about security and how to use Movebank from within R.

Author(s)

Marco Smolla

`show` *Show a Move object*

Description

Displays a summary of a Move object.

Usage

```
  ## S4 method for signature 'Move'
show(object)
  ## S4 method for signature 'MoveStack'
show(object)
```

Arguments

`object`	a Move or MoveStack object

Details

The show function displays a summary of a Move object. This includes:

- animal ID
- species name
- study name
- number of track points
- receiver type
- projection method
- date of file creation
- the first three lines of the spatial data frame
- study citation
- data license
- number of omitted locations due to NAs in the dataset

If the imported data are not from the Movebank database Animal, Species, nPoints, Receiver, and Study are not shown.

Author(s)

Marco Smolla

---

split *Splitting a MoveStack*

Description

Splitting MoveStack in a list of Move objects

Usage

```
  ## S4 method for signature 'MoveStack,missing'
split(x, f, drop=FALSE, ...)
```

Arguments

`x`	MoveStack object
`f`	not needed
`drop`	not needed
`...`	not needed

Details

A MoveStack is splitted into a list of Move objects by the trackIds of the given MoveStack. To stack this list again moveStack can be used.

Author(s)

Marco Smolla

Examples

```
##we would need a moveStack object right here
```

`spTransform` *Transform Move object projection method*

Description

The spTransform function transforms the coordinates stored in the Move object from the default longlat coordinates to the default aequd projection method or a different projection.

Usage

```
  ## S4 method for signature 'Move,character'
spTransform(x,CRSobj,center=FALSE)
  ## S4 method for signature 'Move,missing'
spTransform(x,center=FALSE,...)
```

Arguments

x	a Move or a MoveStack object
CRSobj	a CRS like character that describes the projection method to which the coordinates should be transformed, if missing "+proj=aequd" is used as default value
center	logical, if TRUE the center of the coordinate system is the center of the track; FALSE is default
...	for additional arguments

Details

The spTransform function transforms the coordinates of a Move object by default from "+proj=longlat" to "+proj=aeqd". In this format the coordinates can be used by the brownian.bridge.dyn function.
If center is TRUE the center of the coordinate system is set to the center of the track.

Author(s)

Marco Smolla

Examples

```
## create a Move object
data <- move(system.file("extdata","leroy.csv",package="move"),proj=CRS("+proj=longlat"))

## transform the Move object by default into "+aeqd" projection method
## and center the coordinate system
data2 <- spTransform(data, center=TRUE)

## transform the Move object into another projection method, like mollweide
data3 <- spTransform(data, CRSobj="+proj=moll")

##check projection method
```

```
proj4string(data)
"[1]  +proj=longlat +ellps=WGS84"
proj4string(data2)
"[1]   +proj=longlat +lon_0=-73.8871629 +lat_0=42.73884025 +ellps=WGS84"
proj4string(data3)
"[1]  +proj=moll +ellps=WGS84"
```

subset-method *Returns a single object from a MoveStack*

Description

Returning a single Move object

Usage

```
  ## S4 method for signature 'MoveStack,ANY,ANY'
x[i]
```

Arguments

x	MoveStack object
i	numeric, number of position in the stack

Author(s)

Marco Smolla

summary *A summary of a DBBMM or Move object*

Description

Summarizes the information of the raster from a DBBMM or the data from a Move object

Usage

```
  ## S4 method for signature 'DBBMM'
summary(object)
  ## S4 method for signature 'Move'
summary(object)
```

Arguments

object	a DBBMM or Move object

Details

Returns the projection, extent, and maximum and minimum values of the raster stored within the DBBMM object.

Author(s)

Marco Smolla

`time.lag` *Calculates the time lags between the coordinates*

Description

The time.lag function calculates the time lags between locations.

Usage

```
  ## S4 method for signature 'Move'
time.lag(x,...)
```

Arguments

`x`	a Move object
`...`	further arguments

Details

Optionally the argument units can be passed on to ensure the time lag in a certain unit. For more information on the units argument see the help of difftime.

Value

The function returns numeric vector time lags.

Author(s)

Marco Smolla

Index