

Package ‘GeoLight’

July 2, 2014

Type Package

Title Analysis of light based geolocator data

Version 1.03

Date 2012-12-19

Author Simeon Lisovski, Silke Bauer, Tamara Emmenegger

Maintainer Simeon Lisovski <migration@vogelwarte.ch>

Description The package provides basic functions for global positioning based on light intensity measurements over time. Positioning process includes the determination of sun events, a discrimination of residency and movement periods, the calibration of period-specific data and, finally, the calculation of positions.

License GPL (>= 2)

Depends R (>= 2.10), maps, changepoint

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2013-04-04 20:35:28

R topics documented:

GeoLight-package	2
calib*	5
changeLight	6
coord	7
distanceFilter	9
getElevation	10
gleTrans	11

glfTrans	12
HillEkstromCalib	12
hoopoe1	14
hoopoe2	14
lightFilter	15
loessFilter	16
luxTrans	17
schedule	18
siteMap	18
trip2kml	20
tripMap	21
twilightCalc	22

Index	24
--------------	-----------

GeoLight-package	<i>The GeoLight Package</i>
------------------	-----------------------------

Description

This is a summary of all features of GeoLight, a R-package for analyzing light based geolocator data.

Details

GeoLight is a package to derive geographical positions from daily light intensity pattern. Positioning and calibration methods are based on the threshold-method (Ekstrom 2004, Lisovski *et al.* 2012). A changepoint model from the R package changepoint is implemented to distinguish between periods of residency and movement based on the sunrise and sunset times. Mapping functions are implemented using the R package maps.

Getting Started

We refrain from giving detailed background on the (several steps of) analysis of light-based geolocator data here but strongly recommend the key-publications below.

Updates

We advise all users to update their installation of GeoLight regularly.

Type `news(package="GeoLight")` to read news documentation about changes to the recent and all previous version of the package

Important Notes

Most functions in GeoLight require the same initial units and mostly the format and object type is mandatory:

tFirst	yyyy-mm-dd hh:mm "UTC" (see: as.POSIXct , time zones)
tSecond	as <i>tFirst</i> (e.g. 2008-12-01 17:30)
type	either 1 or 2 depending on whether <i>tFirst</i> is sunrise (1) or sunset (2)
coord	SpatialPoints or a matrix, containing x and y coordinates (in that order)
degElevation	a vector or a single value of sun elevation angle(s) in degrees (e.g. -6)

FUNCTIONS AND DATASETS

In the following, we give a summary of the main functions and sample datasets in the GeoLight package. Alternatively a list of all functions and datasets in alphabetical order is available by typing `library(help=GeoLight)`. For further information on any of these functions, type `help(function name)`.

CONTENTS:

- I. Determination of sunset and sunrise
- II. Residency analysis
- III. Calibration
- IV. Positioning
- V. Data visualisation
- VI. Examples

I. Determination of sunset and sunrise

<code>gleTrans</code>	transformation of already defined twilight events*
<code>glfTrans</code>	transformation of light intensity measurements over time*
<code>luxTrans</code>	transformation of light intensity measurements over time**
<code>lightFilter</code>	filter to remove noise in light intensity measurements during the night
<code>twilightCalc</code>	definition of twilight events (<i>sunrise</i> , <i>sunset</i>) from light intensity measurements

* written for data recorded by geolocator devices from the **Swiss Ornithological Institute** ** written for data recorded by geolocator devices from **Migrate Technology Ltd**

II. Residency Analysis

<code>changeLight</code>	function to distinguish between residency and movement periods
<code>schedule</code>	function to produce a data frame summarizing the residency and movement pattern

III. Calibration

See Lisovski *et al.* 2012 for all implemented calibration methods.

`getElevation` function to calculate the sun elevation angle for data with known position
`HillEkstromCalib` *Hill-Ekstrom calibration* for one or more defined stationary periods

IV. Positioning

`coord` main function to derive a matrix of spatial coordinates
`distanceFilter` filter function to reduce unrealistic positions (not recommended, since the filtering ignore positioning e
`loessFilter` filter function to define outliers in sunrise and sunset times (defined twilight events)

V. Data visualisation

`tripMap` function to map the derived positions and combine the coordinates in time order
`siteMap` function to show the results of the residency analysis on a map

IV. Examples

`calib1` data for calibration: light intensities
`calib2` data for calibration: Calculated twilight events (from `calib1` by `twilightCalc`)
`hoopoe1` light intensity measurements over time recorded on a migratory bird
`hoopoe2` sunrise and sunset times: From light intensity measurement (from `hoopoe1`)

R Packages for Further Spatial Analyses

`spatstat`
`adehabitat`
`gstat`
`trip`
`tripEstimation`
`move`
 ...

Acknowledgements

Steffen Hahn, Felix Liechti, Fraenzi Korner-Nievergelt, Andrea Koelzsch, Eldar Rakhimberdiev, Michael Sumner, Erich Baechler, Eli Bridge, Andrew Parnell, Richard Inger

Authors

Simeon Lisovski, Silke Bauer, Tamara Emmenegger

Maintainer: Simeon Lisovski: <<migration@vogelwarte.ch>>

Discussion Forum: <http://simeonlisovski.wordpress.com/geolight/>

References

- Ekstrom, P.A. (2004) An advance in geolocation by light. *Memoirs of the National Institute of Polar Research*, Special Issue, **58**, 210-226.
- Fudickar, A.M., Wikelski, M., Partecke, J. (2011) Tracking migratory songbirds: accuracy of light-level loggers (geolocators) in forest habitats. *Methods in Ecology and Evolution*, DOI: 10.1111/j.2041-210X.2011.00136.x.
- Hill, C. & Braun, M.J. (2001) Geolocation by light level - the next step: Latitude. *Electronic Tagging and Tracking in Marine Fisheries* (eds J.R. Sibert & J. Nielsen), pp. 315-330. Kluwer Academic Publishers, The Netherlands.
- Hill, R.D. (1994) Theory of geolocation by light levels. *Elephant Seals: Population Ecology, Behavior, and Physiology* (eds L. Boeuf, J. Burney & R.M. Laws), pp. 228-237. University of California Press, Berkeley.
- Lisovski, S. and Hahn, S. (2012) GeoLight - processing and analysing light-based geocator data in R. *Methods in Ecology and Evolution*, doi: 10.1111/j.2041-210X.2012.00248.x
- Lisovski, S., Hewson, C.M., Klaassen, R.H.G., Korner-Nievergelt, F., Kristensen, M.W & Hahn, S. (2012) Geolocation by light: Accuracy and precision affected by environmental factors. *Methods in Ecology and Evolution*, doi: 10.1111/j.2041-210X.2012.00185.x
- Wilson, R.P., Ducamp, J.J., Rees, G., Culik, B.M. & Niekamp, K. (1992) Estimation of location: global coverage using light intensity. *Wildlife telemetry: remote monitoring and tracking of animals* (eds I.M. Priede & S.M. Swift), pp. 131-134. Ellis Horward, Chichester.

calib*

Example data for calibration: Light intensities and twilight events

Description

Light intensity measurements over time (calib1) recorded at the rooftop of the Swiss Ornithological Institute (Lon: 8.0, Lat: 47.01). Defined twilight events from calib1 (calib2). These data serve as an example for calculating the sun elevation angle of an additional data set, which is subsequently used to calibrate the focal dataset.

Usage

```
data(calib1)
```

```
data(calib2)
```

References

Lisovski, S., Hewson, C.M, Klaassen, R.H.G., Korner-Nievergelt, F., Kristensen, M.W & Hahn, S. (2012) Geolocation by light: Accuracy and precision affected by environmental factors. *Methods in Ecology and Evolution*, DOI: 10.1111/j.2041-210X.2012.00185.x.

Examples

```
data(calib2)
attach(calib2)
getElevation(tFirst, tSecond, type, c(8, 47.01))
```

changeLight

Residency analysis using a changepoint model

Description

Function to discriminate between periods of residency and movement based on consecutive sunrise and sunset data. The calculation is based on a changepoint model (**R** Package [changepoint: binseg.mean.cusum](#)) to find multiple changepoints within the data.

Usage

```
changeLight(tFirst, tSecond, type, quantile=0.6, rise.prob=NA, set.prob=NA, days=5, plot=TRUE, summary=TRUE)
```

Arguments

tFirst	date and time of sunrise/sunset (e.g. 2008-12-01 08:30)
tSecond	date and time of sunrise/sunset (e.g. 2008-12-01 17:30)
type	either 1 or 2, defining tFirst as sunrise or sunset respectively
quantile	probability threshold for stationary site selection. Higher values (above the defined quantile of all probabilities) will be considered as changes in the behavior. If specified, rise.prob and/or set.prob will not be considered.
rise.prob	the probability threshold for sunrise : greater or equal values indicates changes in the behaviour of the individual. If, NA sunrise will not be considered.
set.prob	the probability threshold for sunset : higher and equal values indicates changes in the behaviour of the individual. If, NA sunrise will not be considered.
days	a threshold for the length of stationary period. Periods smaller than "days" will not be considered as a residency period
plot	logical, if TRUE a plot will be produced
summary	logical, if TRUE a summary of the results will be printed

Details

The `binseg.mean.cusum` from the `codeR` Package `changePoint` is a function to find a multiple changes in mean for data where no assumption is made on their distribution. The value returned is the result of finding the optimal location of up to `Q` changepoints (in this case as many as possible) using the cumulative sums test statistic.

Value

A list with probabilities for *sunrise* and *sunset* the user settings of the probabilities and the resulting stationary periods given as a vector, with the residency sites as positiv numbers in ascending order (0 indicate movement/migration).

Note

The sunrise and/or sunset times shown in the graph (if `plot=TRUE`) represent hours of the day. However if one or both of the twilight events cross midnight during the recording period the values will be transformed to avoid discontinuity.

Author(s)

Simeon Lisovski & Tamara Emmenegger

References

- Taylor, Wayne A. (2000) Change-Point Analysis: A Powerful New Tool For Detecting Changes.
M. Csorgo, L. Horvath (1997) Limit Theorems in Change-Point Analysis. *Wiley*.
Chen, J. and Gupta, A. K. (2000) Parametric statistical change point analysis. *Birkhauser*.

See Also

[changePoint](#), [binseg.mean.cusum](#)

Examples

```
data(hoopoe2)
attach(hoopoe2)
residency <- changeLight(tFirst, tSecond, type, quantile=0.9)
```

coord

Threshold based geographical positioning

Description

Calculate the latitude and longitude from two subsequent twilight events (sunrise/sunset)

Usage

```
coord(tFirst, tSecond, type, degElevation=-6, site=FALSE, sites, note=TRUE)
```

Arguments

tFirst	date and time of sunrise/sunset (e.g. 2008-12-01 08:30)
tSecond	date and time of sunrise/sunset (e.g. 2008-12-01 17:30)
type	either 1 or 2, defining tFirst as sunrise or sunset respectively
degElevation	sun elevation angle in degrees (e.g. -6 for "civil twilight"). Either a single value, a vector with the same length as tFirst. If site=TRUE a vector with a sun elevation angle for each site in numerical order of the sites (incl. 0 for movements).
site	logical, if TRUE the a sun elevation angle for each site (incl. 0 dor movements) will be used for positioning
sites	a numerical vector assigning each row to a particular period. Stationary periods in numerical order and values >0, migration/movement periods 0.
note	logical, if TRUE a notation of how many positions could be calculated in proportion to the number of failures will be printed at the end.

Details

The format (date and time) of *tFirst* and *tSecond* has to be "yyyy-mm-dd hh:mm" corresponding to Universal Time Zone UTC (see: [as.POSIXct](#), [time zones](#)).

Value

A matrix of coordinates in decimal degrees. First column are longitudes, expressed in degrees east of Greenwich. Second column contains the latitudes in degrees north the equator. If latitude can not be calculated (e.g. during equinox, at high latitudes) the function will return NA for the date.

Author(s)

Simeon Lisovski

References

Montenbruck, O. & Pfleger, T. (2000) Astronomy on the Personal Computer. *Springer*, Berlin.

Examples

```
data(hoopoe2)
attach(hoopoe2)
coord <- coord(tFirst,tSecond,type,degElevation=-6)
## tripMap(coord,xlim=c(-20,20),ylim=c(5,50), main="hoopoe2")
```

distanceFilter *Filter for unrealistic positions within a track*

Description

The filter identifies unrealistic positions. The maximal distance per hour/day can be set corresponding to the particular species.

Usage

```
distanceFilter(tFirst,tSecond,type,degElevation,distance,units)
```

Arguments

tFirst	date and time of sunrise/sunset (e.g. 2008-12-01 08:30)
tSecond	date and time of sunrise/sunset (e.g. 2008-12-01 17:30)
type	either 1 or 2, defining tFirst as sunrise or sunset respectively
degElevation	sun elevation angle in degrees (e.g. -6 for "civil twilight")
distance	the maximal distance in km per units. Distances above will be considered as unrealistic.
units	the time unite corresponding to the distance. Default is "hour", alternative option is "day".

Details

The filter currently depends to some extent on the calibration (degElevation).

Value

Logical vector matching positions that pass the filter.

Author(s)

Simeon Lisovski, Fraenzi Korner-Nievergelt

Examples

```
data(hoopoe2)
attach(hoopoe2)
coord <- coord(tFirst,tSecond,type)
filter <- distanceFilter(tFirst,tSecond,type,distance=30)
tripMap(coord[filter,],xlim=c(-20,20),ylim=c(0,60),main="hoopoe2 (filter)")
```

getElevation	<i>Calculate the appropriate sun elevation angle for known location</i>
--------------	---

Description

Function to calculate the sun elevation angle for light measurements at a known location

Usage

```
getElevation(tFirst, tSecond, type, known.coord, plot=TRUE)
```

Arguments

tFirst	date and time of sunrise/sunset (e.g. 2008-12-01 08:30)
tSecond	date and time of sunrise/sunset (e.g. 2008-12-01 17:30)
type	either 1 or 2, defining tFirst as sunrise or sunset respectively
known.coord	a SpatialPoint or matrix object, containing known x and y coordinates (in that order) for the selected measurement period.
plot	logical, if TRUE a plot will be produced.

Author(s)

Simeon Lisovski

References

Lisovski, S., Hewson, C.M, Klaassen, R.H.G., Korner-Nievergelt, F., Kristensen, M.W & Hahn, S. (2012) Geolocation by light: Accuracy and precision affected by environmental factors. *Methods in Ecology and Evolution*, DOI: 10.1111/j.2041-210X.2012.00185.x.

Examples

```
data(calib2)
attach(calib2)
known.coord <- c(7.1, 46.3)
getElevation(tFirst, tSecond, type, known.coord)
```

gleTrans	<i>Transformation of *.gle files</i>
----------	--------------------------------------

Description

Function to transform *.gle files derived by the software GeoLocator for further analyses in GeoLight.

Usage

```
gleTrans(file)
```

Arguments

`file` the full patch and filename with suffix of the *.gle file.

Details

The *.gle file derived by the software "GeoLocator" (Swiss Ornithological Institute) is a table with interpolated light intensities over time gathered from the *.glf file. Furthermore a column defines whether the light intensity passes the defined light intensity threshold in the morning (sunrise) or in the evening (sunset). This information is used in `gleTrans()`, to create a table with two subsequent twilight events (*tFirst*, *tSecond*) and *type* defining whether *tFirst* refers to sunrise (1) or sunset (2). Date and time information will be transferred into a GeoLight appropriate format (see: [as.POSIXct](#)).

Value

A data.frame suitable for further use in GeoLight.

Author(s)

Simeon Lisovski

See Also

[glfTrans](#)

glfTrans	<i>Transformation of *.glf files</i>
----------	--------------------------------------

Description

Transform *.glf files derived by the software GeoLocator for further analyses in GeoLight.

Usage

```
glfTrans(file)
```

Arguments

file the full patch and filename with suffix of the *.glf file.

Details

The *.glf files produced by the software GeoLocator (Swiss Ornithological Institute) is a table with light intensity measurements over time. glfTrans produces a table with these measurements and transfer the data and time information into the format required by GeoLight format (see: [as.POSIXct](#)).

Value

A data.frame suitable for further use in GeoLight.

Author(s)

Simeon Lisovski

See Also

[gleTrans](#); [luxTrans](#) for transforming *.lux files produced by *Migrate Technology Ltd*

HillEkstromCalib	<i>Hill-Ekstrom calibration</i>
------------------	---------------------------------

Description

Hill-Ekstrom calibration for one or multiple stationary periods.

Usage

```
HillEkstromCalib(tFirst,tSecond,type,site,  
start.angle=-6,distanceFilter=FALSE,  
distance,plot=TRUE)
```

Arguments

tFirst	date and time of sunrise/sunset (e.g. 2008-12-01 08:30)
tSecond	date and time of sunrise/sunset (e.g. 2008-12-01 17:30)
type	either 1 or 2, defining tFirst as sunrise or sunset respectively
site	a numerical vector assigning each row to a particular period. Stationary periods in numerical order and values >0, migration/movement periods 0
start.angle	a single sun elevation angle. The combined process of checking for minimal variance in resulting latitude, which is the initial value for the sun elevation angle in the iterative process of identifying the latitudes with the least variance
distanceFilter	logical, if TRUE the <code>distanceFilter</code> will be used to filter unrealistic positions
distance	if distanceFilter is set TRUE a threshold distance in km has to be set (see: <code>distanceFilter</code>)
plot	logical, if TRUE the function will give a plot with all relevant information

Details

The *Hill-Ekstrom calibration* has been suggested by Hill & Braun (2001) and Ekstrom (2004), and allows for calibrating data during stationary periods at unknown latitudinal positions. The Hill-Ekstrom calibration bases on an increasing error range in latitudes with an increasing mismatch between light level threshold and the used sun angle. This error is strongly amplified with proximity to the equinox times due to decreasing slope of day length variation with latitude. Furthermore, the sign of the error switches at the equinox, i.e. latitude is overestimated before the equinox and underestimated after the equinox (or vice versa depending on autumnal/vernal equinox, hemisphere, and sign of the mismatch between light level threshold and sun angle). When calculating the positions of a stationary period, the variance in latitude is minimal if the sun elevation angle fits to the defined light level threshold. Moreover, the accuracy of positions increases with decreasing variance in latitudes. **However, the method is only applicable for stationary periods and under stable shading intensities.** The plot produced by the function may help to judge visually if the calculated sun elevation angles are realistic (e.g. site 2 in the example below) or not (e.g. site 3 in the example below).

Value

A vector of sun elevation angles corresponding to the Hill-Ekstrom calibration for each defined period.

Author(s)

Simeon Lisovski

References

- Ekstrom, P.A. (2004) An advance in geolocation by light. *Memoirs of the National Institute of Polar Research*, Special Issue, **58**, 210-226.
- Hill, C. & Braun, M.J. (2001) Geolocation by light level - the next step: Latitude. In: *Electronic Tagging and Tracking in Marine Fisheries* (eds J.R. Sibert & J. Nielsen), pp. 315-330. Kluwer Academic Publishers, The Netherlands.

Lisovski, S., Hewson, C.M., Klaassen, R.H.G., Korner-Nievergelt, F., Kristensen, M.W & Hahn, S. (2012) Geolocation by light: Accuracy and precision affected by environmental factors. *Methods in Ecology and Evolution*, DOI: 10.1111/j.2041-210X.2012.00185.x.

Examples

```
data(hoopoe2)
attach(hoopoe2)
residency <- changeLight(tFirst, tSecond, type, rise.prob=0.1, set.prob=0.1, plot=FALSE, summary=FALSE)
HillEkstromCalib(tFirst, tSecond, type, residency$site, -6)
```

hoopoe1	<i>Light intensity measurements over time recorded on a migratory bird</i>
---------	--

Description

Sunlight intensity measurements over time recorded during the first part of the annual migration of a European Hoopoe (*Upupa epops*). All dates/times are measured in Universal Time Zone (UTC).

Usage

```
data(hoopoe1)
```

Format

A table with 24474 rows and 2 columns, rows corresponding to light measurements recorded in ten-minute intervals (datetime).

Source

Baechler, E., Hahn, S., Schaub, M., Arlettaz, R., Jenni, L., Fox, J.W., Afanasyev, V. & Liechti, F. (2010) Year-Round Tracking of Small Trans-Saharan Migrants Using Light-Level Geolocators. *Plos One*, **5**.

hoopoe2	<i>Sunrise and sunset times: From light intensity measurement (hoopoe1)</i>
---------	---

Description

Sunrise and sunset times derived from light intensity measurements over time ([hoopoe1](#)). The light measurements corresponding to the first part of the annual migration of a European Hoopoe (*Upupa epops*).

Usage

```
data(hoopoe1)
```

Format

A table with 340 rows and 3 columns. Each row corresponds to subsequent twilight events ("tFirst" and "tSecond"). The third column ("type") indicates whether the first event is sunrise (1) or sunset (2). All dates/times are measured in Universal Time Zone (UTC).

Source

Baechler, E., Hahn, S., Schaub, M., Arlettaz, R., Jenni, L., Fox, J.W., Afanasyev, V. & Liechti, F. (2010) Year-Round Tracking of Small Trans-Saharan Migrants Using Light-Level Geolocators. *Plos One*, **5**.

Examples

```
data(hoopoe2)
attach(hoopoe2)
coord <- coord(tFirst,tSecond,type,degElevation=-6)
## plot in a map using package maps
# par(oma=c(5,0,0,0))
# map(xlim=c(-20,40),ylim=c(-10,60),interior=F,col="darkgrey")
# map(xlim=c(-20,40),ylim=c(-10,60),boundary=F,lty=2,col="darkgrey",add=T)
# mtext(c("Longitude (degrees)","Latitude (degrees)"),side=c(1,2),line=c(2.2,2.5),font=3)
# map.axes()
# points(coord,col="brown",cex=0.5,pch=20)
```

lightFilter

Filter to remove noise in light intensity measurements during the night

Description

The filter identifies and removes light intensities oscillating around the baseline or few light intensities resulting in a short light peak during the night. Such noise during the night will increase the calculated twilight events using the function `twilightCalc` and therewith the manual work to remove these false twilight events.

Usage

```
lightFilter(light, baseline=NULL, iter=2)
```

Arguments

light	numerical value of the light intensity (usually arbitrary units).
baseline	the light intensity baseline (no light). If Default, it will be calculated as the most frequent value below the mean light intensities.
iter	a numerical value, specifying how many iterations should be computed (see details).

Details

The filter searches for light levels above the baseline and compares the prior and posterior levels. If these values are below the threshold the particular light level will be reduced to the baseline. A few (usually two) iterations might be enough to remove most noise during the night (however, not if such noise occurs at the beginning or at the end where not enough prior or posterior values are available).

Value

numerical vector with the new light levels. Same length as the initial light vector.

Author(s)

Simeon Lisovski

Examples

```
night <- rep(0,50); night[runif(4,0,50)] <- 10; night[runif(4,0,50)] <- -5
nightday <- c(night,rep(30,50))
plot(nightday,type="l",ylim=c(-5,30),ylab="light level",xlab="time (time)")
light2 <- lightFilter(nightday, baseline=0, iter=4)
lines(light2,col="red")
legend("bottomright",c("before","after"),lty=c(1,1),col=c("black","red"),bty="n")
```

loessFilter

Filter to remove outliers in defined sunrise and sunset times

Description

This filter defines outliers based on residuals from a local polynomial regression fitting process ([loess](#)).

Usage

```
loessFilter(tFirst, tSecond, type, k=3, plot=TRUE)
```

Arguments

tFirst	date and time of sunrise/sunset (e.g. 2008-12-01 08:30)
tSecond	date and time of sunrise/sunset (e.g. 2008-12-01 17:30)
type	either 1 or 2, defining tFirst as sunrise or sunset respectively
k	a measure of how many interquartile ranges to take before saying that a particular twilight event is an outlier
plot	codellogical, if TRUE a plot indicating the filtered times will be produced.

Value

Logical vector matching positions that pass the filter.

Author(s)

Simeon Lisovski & Eldar Rakhimberdiev

luxTrans	<i>Transformation of *.lux files</i>
----------	--------------------------------------

Description

Transform *.lux files derived from *Migrate Technology Ltd* geolocator device for further analyses in GeoLight.

Usage

```
luxTrans(file)
```

Arguments

`file` the full path and filename with suffix of the *.lux file.

Details

The *.lux files produced by *Migrate Technology Ltd* are tables with light intensity measurements over time. luxTrans produces a table with these measurements and transfers the data and time information into the format required by GeoLight format (see: [as.POSIXct](#)).

Value

A data frame suitable for further use in GeoLight.

Author(s)

Simeon Lisovski

See Also

[gleTrans](#) for transforming *.glf files produced by the software GeoLocator (*Swiss Ornithological Institute*)

schedule	<i>Summary of migration/movement pattern</i>
----------	--

Description

Function for making a data frame summarising residency and movement pattern.

Usage

```
schedule(tFirst, tSecond, site)
```

Arguments

tFirst	date and time of sunrise/sunset (e.g. 2008-12-01 08:30)
tSecond	date and time of sunrise/sunset (e.g. 2008-12-01 17:30)
site	a vector, indicating the residency period of a particular day (see output: changeLight)

Value

A data.frame with end and start date (yyyy-mm-dd hh:mm, UTC) for each stationary period.

Author(s)

Simeon Lisovski

Examples

```
data(hoopoe2)
attach(hoopoe2)
residency <- changeLight(tFirst, tSecond, type, rise.prob=0.1, set.prob=0.1, plot=FALSE, summary=FALSE)
schedule(tFirst, tSecond, residency$site)
```

siteMap	<i>Draws sites of residency and adds a convex hull</i>
---------	--

Description

Draw a map (from the R Package maps) showing the defined stationary sites

Usage

```
siteMap(coord, site, points=TRUE,
map.range=c("EuroAfrica", "AustralAsia", "America", "World"),
xlim, ylim, xlab="Longitude", ylab="Latitude", lwd=1, lty=1, pch=1, cex=1, col, main, add=FALSE)
```

Arguments

coord	a SpatialPoints or matrix object, containing x and y coordinates (in that order).
site	a numerical vector assigning each row to a particular period. Stationary periods in numerical order and values >0, migration/movement periods 0.
points	logical; if TRUE, the points of each site will also be plotted.
map.range	some possibilities to choose defined areas ("World (default)", "EuroAfrica", "America", "AustralAsia").
xlim	two element numeric vector giving a range of longitudes, expressed in degrees, to which the map is restricted. Longitude is measured in degrees east of Greenwich, so that, in particular, locations in Switzerland have positive longitude. If map.range is defined this argument will not be considered.
ylim	two element numeric vector giving a range of latitudes, expressed in degrees, to which the map is restricted. Latitude is measured in degrees north of the equator, so that, in particular, locations in Switzerland have positive latitude. If map.range is defined this argument will not be considered.
xlab	a title for the x axis.
ylab	a title for the y axis.
lwd	The line width, a positive number, defaulting to 1.
lty	The line type. Line types can either be specified as an integer (0=blank, 1=solid (default), 2=dashed, 3=dotted, 4=dotdash, 5=longdash, 6=twodash) or as one of the character strings "blank", "solid", "dashed", "dotted", "dotdash", "longdash", or "twodash", where "blank" uses "invisible lines" (i.e., does not draw them).
pch	Either an integer specifying a symbol or a single character to be used as the default in plotting points. See points for possible values and their interpretation. Note that only integers and single-character strings can be set as a graphics parameter (and not NA nor NULL).
cex	A numerical value giving the amount by which plotting symbols should be magnified relative to the default.
col	a vector of colors with the same length as the number of defined sites (if Default a predefined color ramp will be used).
main	map title.
add	logical; if TRUE, positions will be added to an existing plot.

Author(s)

Simeon Lisovski

Examples

```
data(hoopoe2)
attach(hoopoe2)
coord <- coord(tFirst,tSecond,type,-6)
filter <- distanceFilter(tFirst,tSecond,type,distance=30)
site <- changeLight(tFirst,tSecond,type,rise.prob=0.1,set.prob=0.1,plot=FALSE,summary=FALSE)$site
siteMap(coord[filter,],site[filter],xlim=c(-20,20),ylim=c(0,60),lwd=2,pch=20,cex=0.5,main="hoopoe2")
```

trip2kml

*Write a file which plots a trip in Google Earth***Description**

This function creates a .kml file from light intensity measurements over time that can be viewed as a trip in Google Earth.

Usage

```
trip2kml(file, tFirst, tSecond, type, degElevation, col.scheme="heat.colors", point.alpha=0.7, cex=1,
```

Arguments

file	A character expression giving the whole path and the name of the resulting output file including the .kml extension.
tFirst	date and time of sunrise/sunset (e.g. 2008-12-01 08:30)
tSecond	date and time of sunrise/sunset (e.g. 2008-12-01 17:30)
type	either 1 or 2, defining tFirst as sunrise or sunset respectively
degElevation	sun elevation angle in degrees (e.g. -6 for "civil twilight"). Either a single value, a vector with the same length as tFirst.
col.scheme	the color scheme used for the points. Possible color schemes are: rainbow , heat.colors , topo.colors , terrain.colors .
point.alpha	a numerical value indicating the transparency of the point colors on a scale from 0 (transparent) to 1 (opaque).
cex	numerical value for the size of the points.
line.col	An character expression (any of colors or hexadecimal notation), or numeric indicating the color of the line connecting the point locations.

Value

This function returns no data. It creates a .kml file in the in the defined path.

Author(s)

Simeon Lisovski and Michael U. Kemp

Examples

```
data(hoopoe2)
attach(hoopoe2)
filter <- distanceFilter(tFirst,tSecond,type,distance=30)
trip2kml("trip.kml", tFirst[filter], tSecond[filter], type[filter],
degElevation=-6, col.scheme="heat.colors", cex=0.7,
line.col="goldenrod")
```

tripMap	<i>Draw the positions and the trip on a map</i>
---------	---

Description

Draw a map (from the R Package maps) with calculated positions connected by a line

Usage

```
tripMap(coord, equinox,
        map.range=c("EuroAfrica", "AustralAsia", "America", "World"),
        xlim, ylim, xlab, ylab, col, main, legend, add)
```

Arguments

coord	a SpatialPoints or matrix object, containing x and y coordinates (in that order).
equinox	logical; if TRUE, the equinox period(s) is shown as a broken blue line.
map.range	some possibilities to choose defined areas (default: "World").
xlim	two element numeric vector giving a range of longitudes, expressed in degrees, to which the map is restricted. Longitude is measured in degrees east of Greenwich, so that, in particular, locations in Switzerland have positive longitude. If map.range is defined this argument will not be considered.
ylim	two element numeric vector giving a range of latitudes, expressed in degrees, to which the map is restricted. Latitude is measured in degrees north of the equator, so that, in particular, locations in Switzerland have positive latitude. If map.range is defined this argument will not be considered.
xlab	a title for the x axis.
ylab	a title for the y axis.
col	colour code or name for the trip line (default equals red).
main	map title
legend	logical; if TRUE, a legend will be added to the plot.
add	logical; if TRUE, positions will be added to an existing plot.

Author(s)

Simeon Lisovski

Examples

```
data(hoopoe2)
attach(hoopoe2)
coord <- coord(tFirst, tSecond, type, -6)
tripMap(coord, xlim=c(-20, 20), ylim=c(0, 60), main="hoopoe2")
```

twilightCalc	<i>Calculate twilight events (sunrise/sunset) from light intensity measurements over time</i>
--------------	---

Description

Defines twilight events (sunrise/sunset) at times when the light intensity measurements (*light*) pass the defined light intensity threshold. An interactive plot can be drawn to assess the calculations and improve e.g. select only the realistic events.

Usage

```
twilightCalc(datetime, light, LightThreshold=TRUE, preSelection=TRUE, maxLight=NULL, ask=TRUE, nsee=500, a
```

Arguments

<code>datetime</code>	date and time of light intensity measurements e.g. 2008-12-01 08:30 "UTC" (see: as.POSIXct, time zones).
<code>light</code>	numerical value of the light intensity (usually arbitrary units).
<code>preSelection</code>	codellogical, if TRUE a pre selection of all calculated twilight events will be offered within the interactive process (<code>ask=TRUE</code>).
<code>LightThreshold</code>	the light intensity threshold for the twilight event calibration. If Default, it will be set slightly above (3 units) the baseline level (measurement during the night).
<code>maxLight</code>	if the geolocator record the maximum light value of a certain time span, give the interval of maximum recordings in minutes (e.g. 5).
<code>ask</code>	logical, if TRUE the interactive plot will start after the calculation.
<code>nsee</code>	number of points to plot per screen.
<code>allTwilights</code>	logical, if TRUE the function returns a list with two tables

Value

if `allTwilights=FALSE`, a data frame. Each row contains two subsequent twilight events (*tFirst*, *tSecond*) and *type* defining wether *tFirst* refers to sunrise (1) or sunset (2). If `allTwilights=TRUE`, a list with the data frame described in the previous sentence and a data frame with all light intensities and a column describing whether each row refers to sunrise (1), sunset (2) or to none of these categories (0).

Note

Depending on shading during light intensity measurements (e.g. due to vegetation, weather, etc., see Lisovski et al. 2012) the light intensities may pass the light intensity threshold several times during the day, resulting false sunrises and sunsets. It is highly recommended to check the derived events visually (`ask=TRUE`). Twilight events can be deleted and undeleted by clicking the (first) mouse button at the particular position in the graph. The second mouse button (or esc) moves the time series forward. Note, that a backward option is not included.

Author(s)

Simeon Lisovski

Index

as.POSIXct, [3](#), [8](#), [11](#), [12](#), [17](#), [22](#)

binseg.mean.cusum, [6](#), [7](#)

calib(calib*), [5](#)

calib*, [5](#)

calib1, [4](#)

calib1(calib*), [5](#)

calib2, [4](#)

calib2(calib*), [5](#)

changeLight, [3](#), [6](#), [18](#)

changePoint, [6](#), [7](#)

colors, [20](#)

coord, [4](#), [7](#)

distanceFilter, [4](#), [9](#), [13](#)

GeoLight (GeoLight-package), [2](#)

GeoLight-package, [2](#)

getElevation, [4](#), [10](#)

gleTrans, [3](#), [11](#), [12](#), [17](#)

glfTrans, [3](#), [11](#), [12](#)

heat.colors, [20](#)

HillekstromCalib, [4](#), [12](#)

hoopoe1, [4](#), [14](#), [14](#)

hoopoe2, [4](#), [14](#)

lightFilter, [3](#), [15](#)

loess, [16](#)

loessFilter, [4](#), [16](#)

luxTrans, [3](#), [12](#), [17](#)

points, [19](#)

rainbow, [20](#)

schedule, [3](#), [18](#)

siteMap, [4](#), [18](#)

terrain.colors, [20](#)

time zones, [3](#), [8](#), [22](#)

topo.colors, [20](#)

trip2kml, [20](#)

tripMap, [4](#), [21](#)

twilightCalc, [3](#), [4](#), [15](#), [22](#)