Package 'argosfilter'

January 2, 2012

Type Package

Title Argos locations filter

Version 0.62

Date 2010-04-11

Author Carla Freitas

Maintainer Carla Freitas <carla@npolar.no>

Description Functions to filters animal satellite tracking data
obtained from Argos. It is especially indicated for telemetry
studies of marine animals, where Argos locations are predominantly of low-quality.

License GPL (>= 2)

Repository CRAN

Date/Publication 2010-04-11 12:57:47

R topics documented:

argosfilter-package *Argos locations filter*

Description

Functions to filter animal satellite tracking data obtained from Argos. It is especially indicated for telemetry studies of marine animals, where Argos locations are predominantly of low-quality.

Details

Package:	argosfilter
Type:	Package
Version:	0.62
Date:	2010-04-11
License:	GPL (>=2)

Author(s)

Carla Freitas <carla@npolar.no> Maintainer: Carla Freitas <carla@npolar.no>

bearing *Bearing between geographical locations*

Description

Function `bearing` calculates the bearing, in degrees, between two geographical locations. Function `bearingTrack` calculates the bearing between a sequence of locations.

Usage

```
bearing(lat1, lat2, lon1, lon2)
bearingTrack(lat, lon)
```

Arguments

`lat1`	latitude of the first location, in decimal degrees
`lat2`	latitude of the second location, in decimal degrees
`lon1`	longitude of the first location, in decimal degrees
`lon2`	longitude of the second location, in decimal degrees
`lat`	vector of latitudes, in decimal degrees
`lon`	vector of longitudes, in decimal degrees

Details

Bearings are calculated using spherical trigonometry. Formulas are given in Zwillinger (2003).

Value

`bearing` returns the bearing, in degrees, between the first location and the second location. 0 is North. `bearingTrack` returns a vector of bearings between the sequence of locations.

Author(s)

Carla Freitas

References

Zwillinger D. (2003) *Standard Mathematical Tables and Formulae*, 31st edition. Chapman & Hall/CRC, Boca Raton, FL.

See Also

`distance`

Examples

```
# Bearing between two geographical locations:
lat1<-rnorm(1,80)
lat2<-rnorm(1,80)
lon1<-rnorm(1,20)
lon2<-rnorm(1,20)
bearing(lat1,lat2,lon1,lon2)

# Bearing between a sequence of 10 geographical locations:
lat<-rnorm(10,80)
lon<-rnorm(10,20)
bearingTrack(lat,lon)
```

---

dist.next *Internal function for package argosfilter*

Description

Internal function for package `argosfilter`

Details

This function is not to be called by the user.

dist.prev *Internal function for package argosfilter*

Description

Internal function for package argosfilter

Details

This function is not to be called by the user.

distance *Great circle distance between geographical coordinates*

Description

Function distance calculates the distance, in km, between two geographical locations following the great circle route. Function distanceTrack calculates the distance, in km, between a sequence of locations.

Usage

```
distance(lat1, lat2, lon1, lon2)
distanceTrack(lat,lon)
```

Arguments

lat1	latitude of the first location, in decimal degrees
lat2	latitude of the second location, in decimal degrees
lon1	longitude of the first location, in decimal degrees
lon2	longitude of the second location, in decimal degrees
lat	vector of latitudes, in decimal degrees
lon	vector of longitudes, in decimal degrees

Details

Distances are calculated using spherical trigonometry. See details on formulae in Zwillinger (2003).

Value

distance returns the distance between the two locations. distanceTrack returns a vector of distances between the sequence of locations. In both cases, distances are given in km.

Author(s)

Carla Freitas

References

Zwillinger D. (2003) *Standard Mathematical Tables and Formulae*, 31st edition. Chapman & Hall/CRC, Boca Raton, FL.

See Also

bearing

Examples

```
# Distance between two geographical locations
lat1<-rnorm(1,80)
lon1<-rnorm(1,20)
lat2<-rnorm(1,80)
lon2<-rnorm(1,20)
distance(lat1,lat2,lon1,lon2)

# Distance between a sequence of 10 geographical locations:
lat<-rnorm(10,80)
lon<-rnorm(10,20)
distanceTrack(lat,lon)
```

---

distance_m *Internal function for package argosfilter*

Description

Internal function for package argosfilter

Details

This function is not to be called by the user.

---

internal.angles *Internal function for package argosfilter*

Description

Internal function for package argosfilter

Details

This function is not to be called by the user.

radian *Degrees to radians*

Description

Converts degrees to radians

Usage

```
radian(degree)
```

Arguments

degree	vector of values in degrees

Value

Returns a vector of values in radians.

Author(s)

Carla Freitas

Examples

```
# Convert latitudes and longitudes from decimal degrees to radians
lat<-rnorm(10,80)
lon<-rnorm(10,20)
lat_rad<-radian(lat)
lon_rad<-radian(lon)
```

sdafilter *Filter Argos locations*

Description

This function filters location data obtained from Argos, using the Freitas et al. (2008) algorithm.

Usage

```
sdafilter(lat, lon, dtime, lc, vmax = 2, ang = c(15, 25), distlim = c(2500, 5000))
```

Arguments

lat	a numeric vector of latitudes, in decimal degrees
lon	a numeric vector of longitudes, in decimal degrees
dtime	a vector of class POSIXct with date and time for each location
lc	a numeric or character vector of Argos location classes. Argos locations Z can be entered as "Z", "z" or -9
vmax	speed threshold, in m/s. Default is 2 m/s
ang	angles of the spikes to be removed. Default is c(15,25). No spikes are removed if ang=-1
distlim	lengths of the above spikes, in meters. Default is c(2500,5000)

Details

Locations are filtered using the algorithm described in Freitas et al. (2008). The algorithm first removes all locations with location class Z (-9), which are the points for which the location process failed. Then all locations requiring unrealistic swimming speeds are removed, using the McConnell et al. (1992) algorithm, unless the point is located at less than 5 km from the previous location. This procedure enables retaining good quality locations for which high swimming speeds result from location being taken very close to each other in time. The default maximum speed threshold is 2 m/s. The last step is optional, and enables to remove unlikely spikes from the animal's path. The angles of the spikes should be specified in ang, and their respective length in distlim. The default is c(15,25) for ang and c(2500,5000) for distlim, meaning that all spikes with angles smaller than 15 and 25 degrees will be removed if their extension is higher than 2500 m and 5000 m respectively. No spikes are removed if ang=-1. ang and distlim vectors must have the same length.

Value

Returns a vector with the following elements: "removed" (location removed by the filter), "not" (location not removed) and "end_location" (location at the end of the track where the algorithm could not be applied).

Author(s)

Carla Freitas, with contributions from Anne Goarant and Catriona MacLeod

References

Freitas, C., Lydersen, C., Ims, R.A., Fedak, M.A. and Kovacs, K.M. (2008) A simple new algorithm to filter marine mammal Argos locations *Marine Mammal Science* 24:315-325.

McConnell, B.J., Chambers, C. and Fedak, M.A. (1992) Foraging ecology of southern elephant seals in relation to the bathymetry and productivity of the Southern Ocean. *Antarctic Science* 4:393-398.

See Also

vmask

Examples

```
data(seal)
lat<-seal$lat
lon<-seal$lon
dtime<-seal$dtime
lc<-seal$lc

# plot unfiltered data
plot(lon,lat,col="lightgrey",type="l",xlim=c(5,18), ylim=c(77.1,79.1),xlab="Longitude",ylab="Latitude")

# filter by speed only
mfilter<-vmask(lat,lon,dtime,2)
mfilter[1:10]
lines(lon[which(mfilter=="not")],lat[which(mfilter=="not")],col="red")

# filter data using sdafilter
cfilter<-sdafilter(lat, lon, dtime, lc)
cfilter[1:20]
lines(lon[which(cfilter=="not")],lat[which(cfilter=="not")],col="blue")

# check number of locations (by location class) removed by each filter
table(lc,mfilter)
table(lc,cfilter)
```

`seal` *Satellite tracking data from a ringed seal*

Description

Satellite tracking data from a ringed seal tagged in Svalbard, Norway. Data were provided by C. Lydersen and K. M. Kovacs (Norwegian Polar Institute).

Usage

```
data(seal)
```

Format

A data frame with 1561 observations on the following 4 variables.

`dtime` a vector of class POSIXt with date and time for each location

`lat` a numeric vector with latitudes, in decimal degrees

`lon` a numeric vector with longitudes, in decimal degrees

`lc` a numeric vector with location classes LC (-9 refers to LC Z, -2 to LC B and -1 to LC A)

Examples

```
data(seal)
lat<-seal$lat
lon<-seal$lon
# plot unfiltered data
plot(lon,lat)
```

---

vmask *Filter locations for speed*

Description

This function filters location data obtained from Argos, using the MacConnell al. (1992) algorithm.

Usage

```
vmask(lat, lon, dtime, vmax)
```

Arguments

lat	a numeric vector of latitudes, in decimal degrees
lon	a numeric vector of longitudes, in decimal degrees
dtime	a vector of class POSIXct with date and time for each location
vmax	speed threshold, in m/s

Details

Locations are filtered using the algorithm described in MacConnell et al. (1992). Since this algorithm calculates for each location the root mean square (rms) of the speeds to the previous, 2nd previous, next and 2nd next location, high swimming speeds can be obtained for points that are adjacent to outlier locations. Therefore, when the algorithm is applied to a set of locations, only the peaks in rms (that are above the maximum speed) are removed. Other locations are not removed even if above the speed limit. rms is then recalculated n times until all locations are below the speed threshold.

Value

Returns a vector with the following elements: "removed" (location removed by the filter), "not" (location not removed) and "end_location" (location at the end of the track where the algorithm could not be applied).

Author(s)

Carla Freitas

References

McConnell, B.J., Chambers, C. and Fedak, M.A. (1992) Foraging ecology of southern elephant seals in relation to the bathymetry and productivity of the Southern Ocean. *Antarctic Science* 4:393-398.

See Also

sdafilter

Examples

```
data(seal)
lat<-seal$lat
lon<-seal$lon
dtime<-seal$dtime
lc<-seal$lc

# filter by speed
mfilter<-vmask(lat,lon,dtime,2)

# plot unfiltered (grey) and filtered data (green)
plot(lon,lat,col="lightgrey",type="l",xlim=c(5,18), ylim=c(77.1,79.1),xlab="Longitude",ylab="Latitude")
lines(lon[which(mfilter=="not")],lat[which(mfilter=="not")],col="darkgreen")
```

Index