Package ‘SDLfilter’

July 21, 2025

Type Package

Title Filtering and Assessing the Sample Size of Tracking Data

Version 2.3.3

Date 2023-11-07

Author Takahiro Shimada

Maintainer Takahiro Shimada <`taka.shimada@gmail.com`>

Description Functions to filter GPS/Argos locations, as well as assessing the sample size for the analysis of animal distributions. The filters remove temporal and spatial duplicates, fixes located at a given height from estimated high tide line, and locations with high error as described in Shimada et al. (2012) <doi:10.3354/meps09747> and Shimada et al. (2016) <doi:10.1007/s00227-015-2771-0>. Sample size for the analysis of animal distributions can be assessed by the conventional area-based approach or the alternative probability-based approach as described in Shimada et al. (2021) <doi:10.1111/2041-210X.13506>.

Depends R (>= 3.5.0), ggplot2

Imports geosphere, data.table, gridExtra, ggmap, maps, pracma, lubridate, dplyr, emmeans, utils, sf, stars, ggspatial

License GPL-2 | file LICENSE

URL `https://github.com/TakahiroShimada/SDLfilter`

BugReports `https://github.com/TakahiroShimada/SDLfilter/issues`

LazyData true

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Repository CRAN

Date/Publication 2023-11-10 00:00:11 UTC

Contents

`asymptote` *Horizontal asymptotes of rational functions*

Description

Function to find horizontal asymptotes of a rational function.

Usage

```
asymptote(
  data = NULL,
  x = NULL,
  y = NULL,
  degree = "optim",
```

```
    upper.degree = 5,
    d1 = NA,
    d2 = NA,
    threshold = 0.95,
    proportional = TRUE,
    max.asymptote = 1,
    estimator = "glm",
    ci.level = 0.95,
    ...
  )
```

Arguments

`data`	An output object from `boot_overlap`, `combn_overlap`, or `boot_area`.
`x, y`	Numeric vectors of independent (x) and dependent (y) variables. These parameters will be ignored if *data* is supplied.
`degree`	The default 'optim' option selects the maximal degree of numerator and denominator of a rational function that minimises the mean squared error. Alternatively, an integer can be used to specify the maximal degree. The 'optim' option is recommended unless there is a strong reason that a maximal degree should be specified.
`upper.degree`	The upper limit of the maximal degree to be assessed when the 'optim' option is selected. Default is 5, meaning the "optimal" degree is searched from 1 and 10. The default usually gives good results. If the fit does not look good, a larger value may result in a better fit.
`d1, d2`	(Deprecated) Maximal degrees of numerator (d1) and denominator (d2) of a rational function. d1 and d2 must be equal. Use *degree* instead.
`threshold`	Threshold value for considering an asymptote. Once the y value reaches the threshold, it is considered that an asymptote is reached.
`proportional`	If TRUE (default), a threshold is calculated as *estimated asymptote* * *threshold*. If FALSE, the value specified in *threshold* is used in the analysis.
`max.asymptote`	The maximum limit of an expected asymptote. Default is 1 (i.e. maximum probability). If it is unknown, set as NA (e.g. max.asymptote = NA).
`estimator`	Method used to estimate the mean or predicted *y* relative to *x* (e.g. sample size). Available options are 'mean' using arithmetic means and 'glm' using the `glm` function.
`ci.level`	Confidence level for the mean or predicted *y*, which will be used to assess if/when an asymptote has been reached. If NULL, only the mean and predicted *y* are used for the assessment (see details).
`...`	Optional arguments passed to `glm`.

Details

This function fits a rational function to the input data. When an output object from `boot_overlap`, `combn_overlap` or `boot_area` is supplied, a rational function is fit to the means or predicted values of the bootstrap results (e.g. mean overlap probability) as a function of *x* (e.g. sample size). It then estimates horizontal asymptotes and identifies the sample size when an asymptote is considered. If ci.level = NULL and threshold = 0.95, an asymptote is considered when the mean or predicted y value reaches above 95 If ci.level is specified (e.g. 0.95) and threshold = 0.95, an asymptote is considered when the mean or predicted y value AND the confidence interval are above 95 When the "PHR" method was used in `boot_overlap`, binomial is generally a sensible `family` object for the GLM. gaussian and Gamma are often good options when the maximum *y* value exceeds 1 (e.g. area size). Please caution if estimated horizontal asymptote is very different from the expected asymptote. For example, the estimated horizontal asymptote should be around 1 when overlaps between UDs are calculated using the "PHR" method. see `boot_overlap`.

Value

A list containing a data frame (rational function fit associated with x values), an estimated horizontal asymptote, the minimum sample size if an asymptote is reached, and the estimated optimal degree of numerator and denominator of the rational function.

Author(s)

Takahiro Shimada

References

Shimada T, Thums M, Hamann M, Limpus CJ, Hays GC, FitzSimmons N, Wildermann NE, Duarte CD, Meekan MG (2021) Optimising sample sizes for animal distribution analysis using tracking data. *Methods in Ecology and Evolution* 12(2):288-297 doi:10.1111/2041210X.13506

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery (2007). *Numerical Recipes: The Art of Numerical Computing*. Third Edition, Cambridge University Press, New York.

See Also

`boot_overlap`, `combn_overlap`, `boot_area`

`Australia` *A map of Australia*

Description

This map layer outlines the coast of Australia.

Usage

```
Australia
```

Format

A data.frame

`bathymodel` *Bathymetry model for Sandy Strait, Australia*

Description

A high resolution bathymetry model (100 m) for the Sandy Strait region developed by Beaman, R.J. (2010).

Usage

```
bathymodel
```

Format

A stars

Source

https://www.deepreef.org/

References

Beaman, R.J. (2010) Project 3DGBR: A high-resolution depth model for the Great Barrier Reef and Coral Sea. *Marine and Tropical Sciences Research Facility (MTSRF) Project 2.5i.1a Final Report*, MTSRF, Cairns, Australia, pp. 13 plus Appendix 1.

`boot_area` *Cumulative analysis of collective areas by bootstrapping*

Description

Function to calculate collective areas (merged x% Utilisation Distributions or UDs) of *n* individuals by bootstrapping.

Usage

```
boot_area(
  data,
  cell.size = NA,
  R = 1000,
  percent = 50,
  quantiles = c(0.25, 0.5, 0.75)
)
```

Arguments

`data`	A matrix or list of RasterLayer/SpatRaster objects. Each row of the matrix or each RasterLayer/SpatRaster object contains a utilisation distribution (or other statistics that sums to 1 - e.g. proportion of time spent). **The grid size and geographical extent must be consistent across each row of the matrix or each RasterLayer/SpatRaster object.** The function assumes that each column of the matrix is associated with a unique geographical location or that each RasterLayer/SpatRaster has exactly the same geographical extent and resolution.
`cell.size`	A numeric value specifying the grid cell size of the input data in metres.
`R`	An integer specifying the number of iterations. A larger *R* is required when the sample size is large. R > sample size x 100 is recommended (e.g. R > 1000 for a sample size 10).
`percent`	An integer specifying the percent volume of each UD to be considered in the analysis.
`quantiles`	A vector or a number to specify the quantiles to be calculated in the summary of the results.

Details

This function calculates collective areas (e.g. 50% UDs) of 1 to *n* individuals by bootstrapping.

Value

A list containing two data frames - raw results and summary (mean, sd, sem and quantiles at each sample size).

Author(s)

Takahiro Shimada

References

Shimada T, Thums M, Hamann M, Limpus CJ, Hays GC, FitzSimmons N, Wildermann NE, Duarte CD, Meekan MG (2021) Optimising sample sizes for animal distribution analysis using tracking data. *Methods in Ecology and Evolution* 12(2):288-297 doi:10.1111/2041210X.13506

See Also

`boot_overlap`, `combn_overlap`

Examples

```
## Not run:

#1 Utilisation distributions of flatback turtles (n = 15).
data(ud_raster)

#2 Calculate collective areas from 3000 random permutation
area <- boot_area(ud_raster, R = 3000, percent = 50)
```

```
#3 Find the minimum sample size required to estimate the general distribution.
a <- asymptote(area, upper.degree = 10, estimator = 'glm', family = gaussian, max.asymptote = NA)

#4 Plot the mean collective area and rational function fit relative to the sample sizes.
ggplot(data = a$results, aes(x = x))+
  geom_pointrange(aes(y = y, ymin = y_lwr, ymax = y_upr)) +
  geom_point(aes(y = y), size = 2) +
 scale_x_continuous(breaks = seq(0, 15, 3), limits = c(2,15), name = "Animals tracked (n)") +
  scale_y_continuous(name = expression(Area~(km^2)), labels=function(x) x/1e6)

## End(Not run)
```

`boot_overlap` *Bootstrap overlaps between Utilisation Distributions (UDs)*

Description

Function to calculate overlaps between UDs relative to sample size by bootstrapping.

Usage

```
boot_overlap(
  data,
  R = 1000,
  method = "PHR",
  percent = 100,
  quantiles = c(0.25, 0.5, 0.75)
)
```

Arguments

`data`	A matrix or list of RasterLayer/SpatRaster objects. Each row of the matrix or each RasterLayer/SpatRaster object contains a utilisation distribution (or other statistics that sums to 1 - e.g. proportion of time spent). **The grid size and geographical extent must be consistent across each row of the matrix or each RasterLayer/SpatRaster object.** The function assumes that each column of the matrix is associated with a unique geographical location or that each RasterLayer/SpatRaster has exactly the same geographical extent and resolution.
`R`	An integer specifying the number of iterations. A larger *R* is required when the sample size is large. R > sample size x 100 is recommended (e.g. R > 1000 for a sample size 10).
`method`	The overlap quantification method. "HR" is for the proportion of an individual's home range overlapped by the known habitats of other individuals. "PHR" is for the probability of an individual to be within the known habitats of other individuals. "VI", "BA" and "UDOI" quantify overlap between UDs using the full probabilistic properties as described in Fieberg and Kochanny (2005). For

	the latter three options, the function calculates overlaps between each additional UD and a collective UD. To generate a collective UD, each UD is overlaid and averaged at each grid cell so the probability density of the collective UD sums up to 1.
`percent`	An integer specifying the percent volume of each UD to be considered in the analysis.
`quantiles`	A vector or a number to specify the quantiles to be calculated in the summary of the results.

Details

This function calculates and bootstraps overlap between UDs based on the areas ("HR"), areas of collective UDs and the probability distribution of each individual ("PHR"), or the probability distribution of an individual and an averaged probability distribution of collective individuals ("VI", "BA", "UDOI").

Value

A list containing two data frames - raw results and summary (mean, sd, sem and quantiles at each sample size).

Author(s)

Takahiro Shimada

References

Shimada T, Thums M, Hamann M, Limpus CJ, Hays GC, FitzSimmons N, Wildermann NE, Duarte CD, Meekan MG (2021) Optimising sample sizes for animal distribution analysis using tracking data. *Methods in Ecology and Evolution* 12(2):288-297 doi:10.1111/2041210X.13506

Fieberg J & Kochanny CO (2005) Quantifying home-range overlap: The importance of the utilization distribution. *The Journal of Wildlife Management*, 69(4), 1346–1359. doi:10.2193/0022-541x(2005)69[1346:Qhotio]2.0.Co;2

See Also

`combn_overlap`, `boot_area`

Examples

```
## Not run:

#1 Utilisation uistributions of flatback turtles (n = 15).
data(ud_matrix)

#2 Calculate overlap probability from 2000 random permutation.
overlap <- boot_overlap(ud_matrix, R = 2000, method = "PHR")

#3 Find the minimum sample size required to estimate the general distribution.
a <- asymptote(overlap, upper.degree = 10, estimator = 'glm', family = binomial)
```

```
#4 Plot the mean probability and rational function fit relative to the sample sizes.
ggplot(data = a$results, aes(x = x))+
  geom_pointrange(aes(y = y, ymin = y_lwr, ymax = y_upr)) +
  geom_hline(yintercept = a$h.asymptote*0.95, linetype = 2) +
 scale_x_continuous(breaks = seq(0, 15, 3), limits = c(2,15), name = "Animals tracked (n)") +
  scale_y_continuous(limits = c(0.5,1), name = "Overlap probability")

## End(Not run)
```

`combn_overlap` *Quantifying overlaps between all possible combination of Utilisation Distributions (UDs)*

Description

Function to calculate overlaps between all possible combination of UDs relative to sample size.

Usage

```
combn_overlap(
  data,
  method = "PHR",
  percent = 100,
  quantiles = c(0.25, 0.5, 0.75)
)
```

Arguments

`data`	A matrix or list of RasterLayer/SpatRaster objects. Each row of the matrix or each RasterLayer/SpatRaster object contains a utilisation distribution (or other statistics that sums to 1 - e.g. proportion of time spent). **The grid size and geographical extent must be consistent across each row of the matrix or each RasterLayer/SpatRaster object.** The function assumes that each column of the matrix is associated with a unique geographical location or that each RasterLayer/SpatRaster has exactly the same geographical extent and resolution.
`method`	The overlap quantification method. "HR" is for the proportion of an individual's home range overlapped by the known habitats of other individuals. "PHR" is for the probability of an individual to be within the known habitats of other individuals. "VI", "BA" and "UDOI" quantify overlap between UDs using the full probabilistic properties as described in Fieberg and Kochanny (2005). For the latter three options, the function calculates overlaps between each additional UD and a collective UD. To generate a collective UD, each UD is overlaid and averaged at each grid cell so the probability density of the collective UD sums up to 1.
`percent`	An integer specifying the percent volume of each UD to be considered in the analysis.
`quantiles`	A vector or a number to specify the quantiles to be calculated in the summary of the results.

Details

This function calculates overlap between all possible combination of input UDs based on the areas ("HR"), areas of collective UDs and the probability distribution of each individual ("PHR"), or the probability distribution of an individual and an averaged probability distribution of collective individuals ("VI", "BA", "UDOI").

Value

A list containing two data frames - raw results and summary (mean, sd, sem and quantiles at each sample size).

Author(s)

Takahiro Shimada

References

Shimada T, Thums M, Hamann M, Limpus CJ, Hays GC, FitzSimmons N, Wildermann NE, Duarte CD, Meekan MG (2021) Optimising sample sizes for animal distribution analysis using tracking data. *Methods in Ecology and Evolution* 12(2):288-297 doi:10.1111/2041210X.13506

Fieberg J & Kochanny CO (2005) Quantifying home-range overlap: The importance of the utilization distribution. *The Journal of Wildlife Management*, 69(4), 1346–1359. doi:10.2193/0022-541x(2005)69[1346:Qhotio]2.0.Co;2

See Also

`boot_overlap`, `boot_area`

Examples

```
## Not run:

#1 Utilisation uistributions of flatback turtles (n = 15).
data(ud_matrix)

#2 Calculate overlap probability from all combination of the UDs.
overlap <- combn_overlap(ud_matrix, method = "PHR")

#3 Find the minimum sample size required to estimate the general distribution.
a <- asymptote(overlap, upper.degree = 10, ci.level = NULL)

#4 Plot the mean probability and rational function fit relative to the sample sizes.
ggplot(data = a$results, aes(x = x, y = y))+
  geom_point() +
  geom_hline(yintercept = a$h.asymptote*0.95, linetype = 2) +
 scale_x_continuous(breaks = seq(0, 15, 3), limits = c(2,15), name = "Animals tracked (n)") +
  scale_y_continuous(limits = c(0.5,1), name = "Overlap probability")

## End(Not run)
```

`ddfilter` *Filter locations using a data driven filter*

Description

Function to remove locations by a data driven filter as described in Shimada et al. (2012).

Usage

```
ddfilter(sdata, vmax = 8.9, vmaxlp = 1.8, qi = 4, ia = 90, method = 1)
```

Arguments

`sdata`	A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon", "qi". See the data `turtle` for an example. The function filters the input data by a unique "id" (e.g. transmitter number, identifier for each animal). "DateTime" is the GMT date & time of each location in class `POSIXct` or `character` with the following format "2012-06-03 01:33:46". "lat" and "lon" are the latitude and longitude of each location in decimal degrees. "qi" is the quality index associated with each location fix. The input values can be either the number of GPS satellites or Argos Location Classes. Argos Location Classes will be converted to numerical values, where "A", "B", "Z" will be replaced with "-1", "-2", "-3" respectively. The greater number indicates a higher accuracy.
`vmax`	A numeric value specifying a threshold of speed from a previous and/or to a subsequent fix. Default is 8.9km/h. If this value is unknown, it can be estimated from *sdata* using the function `vmax`.
`vmaxlp`	A numeric value specifying a threshold of speed, which is used to evaluate the locations of loop trips. Default is 1.8 km/h. If this value is unknown, it can be estimated from *sdata* using the function `vmaxlp`.
`qi`	An integer specifying a threshold of quality index, which is used to evaluate the locations of loop trips. Default is 4.
`ia`	An integer specifying a threshold of inner angle, which is used to evaluate the locations of loop trips. Default is 90 degrees.
`method`	An integer (1 or 2) specifying how locations should be filtered with *vmax*. Default is 1 (both way) and removes a location if the speed from a previous AND to a subsequent location exceeds *vmax*. Select 2 (one way) to remove a location if the speed from a previous OR to a subsequent location exceeds *vmax*. For the latter, the filter examines successive suspect locations (i.e. the speed from a previous and/or to a subsequent location exceeds *vmax*) and retain one location that is associated with the minimum speed from a previous and/or to a subsequent location.

Details

Locations are removed if the speed from a previous and/or to a subsequent location exceeds *vmax*, or if all of the following criteria apply: the associated quality index is less than or equal to *qi*, the inner angle is less than or equal to *ia* and the speed either from a previous or to a subsequent location exceeds *vmaxlp*. If *vmax* and *vmaxlp* are unknown, they can be estimated using the functions `vmax` and `vmaxlp` respectively.

Value

The input data is returned without locations identified by this filter. The following columns are added: "pTime", "sTime", "pDist", "sDist", "pSpeed", "sSpeed", "inAng". "pTime" and "sTime" are hours from a previous and to a subsequent fix respectively. "pDist" and "sDist" are straight distances in kilometres from a previous and to a subsequent fix respectively. "pSpeed" and "sSpeed" are linear speed from a previous and to a subsequent fix respectively. "inAng" is the degree between the bearings of lines joining successive location points.

Author(s)

Takahiro Shimada

References

Shimada T, Jones R, Limpus C, Hamann M (2012) Improving data retention and home range estimates by data-driven screening. *Marine Ecology Progress Series* 457:171-180 doi:10.3354/meps09747

See Also

`ddfilter_speed`, `ddfilter_loop`, `vmax`, `vmaxlp`

Examples

```
#### Load data sets
## Fastloc GPS data obtained from a green turtle
data(turtle)

## A Map for the example site
data(Australia)
data(SandyStrait)


#### Filter temporal and/or spatial duplicates
turtle.dup <- dupfilter(turtle, step.time=5/60, step.dist=0.001)


#### ddfilter
## Using the built-in function to estimate the threshold speeds
V <- vmax(turtle.dup)
VLP <- vmaxlp(turtle.dup)
turtle.dd <- ddfilter(turtle.dup, vmax=V, vmaxlp=VLP)
```

```
## Or using user specified threshold speeds
turtle.dd <- ddfilter(turtle.dup, vmax=9.9, qi=4, ia=90, vmaxlp=2.0)


#### Plot data removed or retained by ddfilter
## Entire area
p1 <- to_map(turtle.dup, bgmap=Australia, point.size = 2, line.size = 0.5, axes.lab.size = 0,
             multiplot = FALSE, point.bg = "red",
             title.size=15, title="Entire area")[[1]] +
  geom_point(aes(x=lon, y=lat), data=turtle.dd, size=2, fill="yellow", shape=21)+
  geom_point(aes(x=x, y=y), data=data.frame(x=c(154, 154), y=c(-22, -22.5)),
              size=3, fill=c("yellow", "red"), shape=21) +
  annotate("text", x=c(154.3, 154.3), y=c(-22, -22.5), label=c("Retained", "Removed"),
            colour="black", size=4, hjust = 0)

## Zoomed in
p2 <- to_map(turtle.dup, bgmap=SandyStrait, xlim=c(152.7, 153.2), ylim=(c(-25.75, -25.24)),
             axes.lab.size = 0, point.size = 2, point.bg = "red", line.size = 0.5,
             multiplot = FALSE, title.size=15, title="Zoomed in")[[1]] +
geom_path(aes(x=lon, y=lat), data=turtle.dd, linewidth=0.5, colour="black", linetype=1) +
geom_point(aes(x=lon, y=lat), data=turtle.dd, size=2, colour="black", shape=21, fill="yellow")

gridExtra::marrangeGrob(list(p1, p2), nrow=1, ncol=2)
```

`ddfilter_loop` *Filter locations by quality index, inner angle, and speed*

Description

A partial component of `ddfilter`, although works as a stand-alone function. This function removes locations by speed, inner angle, and quality index as described in Shimada et al. (2012).

Usage

```
ddfilter_loop(sdata, qi = 4, ia = 90, vmaxlp = 1.8)
```

Arguments

`sdata`	A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon", "qi". See the data `turtle` for an example. The function filters the input data by a unique "id" (e.g. transmitter number, identifier for each animal). "DateTime" is the GMT date & time of each location in class `POSIXct` or `character` with the following format "2012-06-03 01:33:46". "lat" and "lon" are the latitude and longitude of each location in decimal degrees. "qi" is the quality index associated with each location fix. The input values can be either the number of GPS satellites or Argos Location Classes. Argos Location Classes will be converted to numerical values, where "A", "B", "Z" will be replaced with "-1", "-2", "-3" respectively. The greater number indicates a higher accuracy.

`qi`	An integer specifying a threshold of quality index, which is used to evaluate the locations of loop trips. Default is 4.
`ia`	An integer specifying a threshold of inner angle, which is used to evaluate the locations of loop trips. Default is 90 degrees.
`vmaxlp`	A numeric value specifying a threshold of speed, which is used to evaluate the locations of loop trips. Default is 1.8 km/h. If this value is unknown, it can be estimated from *sdata* using the function `vmaxlp`.

Details

This function removes locations if all of the following criteria apply: the number of source satellites are less than or equal to *qi*, the inner angle is less than and equal to *ia* and the speed either from a previous or to a subsequent location exceeds *vmaxlp*. If *vmaxlp* is unknown, it can be estimated using the function `vmaxlp`.

Value

The input data is returned without locations identified by this filter. The following columns are added: "pTime", "sTime", "pDist", "sDist", "pSpeed", "sSpeed", "inAng". "pTime" and "sTime" are hours from a previous and to a subsequent fix respectively. "pDist" and "sDist" are straight distances in kilometres from a previous and to a subsequent fix respectively. "pSpeed" and "sSpeed" are linear speed from a previous and to a subsequent fix respectively. "inAng" is the degree between the bearings of lines joining successive location points.

Author(s)

Takahiro Shimada

References

Shimada T, Jones R, Limpus C, Hamann M (2012) Improving data retention and home range estimates by data-driven screening. *Marine Ecology Progress Series* 457:171-180 doi:10.3354/meps09747

See Also

`ddfilter`, `ddfilter_speed`, `vmaxlp`

`ddfilter_speed` *Filter locations by speed*

Description

A partial component of `ddfilter`, although works as a stand-alone function. This function removes locations by a given threshold speed as described in Shimada et al. (2012).

Usage

```
ddfilter_speed(sdata, vmax = 8.9, method = 1)
```

Arguments

`sdata`	A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon", "qi". See the data `turtle` for an example. The function filters the input data by a unique "id" (e.g. transmitter number, identifier for each animal). "DateTime" is the GMT date & time of each location in class `POSIXct` or `character` with the following format "2012-06-03 01:33:46". "lat" and "lon" are the latitude and longitude of each location in decimal degrees. "qi" is the quality index associated with each location fix. The input values can be either the number of GPS satellites or Argos Location Classes. Argos Location Classes will be converted to numerical values, where "A", "B", "Z" will be replaced with "-1", "-2", "-3" respectively. The greater number indicates a higher accuracy.
`vmax`	A numeric value specifying a threshold of speed from a previous and/or to a subsequent fix. Default is 8.9km/h. If this value is unknown, it can be estimated from *sdata* using the function `vmax`.
`method`	An integer (1 or 2) specifying how locations should be filtered. Default is 1 and removes a location if the speed from a previous AND to a subsequent location exceeds *vmax*. Select 2 to remove a location if the speed from a previous OR to a subsequent location exceeds *vmax*. For the latter, the filter examines successive suspect locations (i.e. the speed from a previous and/or to a subsequent location exceeds *vmax*) and retain one location that is associated with the minimum speed from a previous and to a subsequent location.

Details

This function removes locations if the speed from a previous and/or to a subsequent location exceeds a given threshold speed. If *vmax* is unknown, it can be estimated using the function `vmax`.

Value

The input data is returned without locations identified by this filter. The following columns are added: "pTime", "sTime", "pDist", "sDist", "pSpeed", "sSpeed". "pTime" and "sTime" are hours from a previous and to a subsequent fix respectively. "pDist" and "sDist" are straight distances in kilometres from a previous and to a subsequent fix respectively. "pSpeed" and "sSpeed" are linear speed from a previous and to a subsequent fix respectively.

Author(s)

Takahiro Shimada

References

Shimada T, Jones R, Limpus C, Hamann M (2012) Improving data retention and home range estimates by data-driven screening. *Marine Ecology Progress Series* 457:171-180 doi:10.3354/meps09747

See Also

`ddfilter`, `ddfilter_loop`, `vmax`, `track_param`

`depthfilter` *Filter locations by water depth*

Description

Function to filter locations according to bathymetry and tide.

Usage

```
depthfilter(
  sdata,
  bathymetry,
  bilinear = TRUE,
  qi = 4,
  tide,
  tidal.plane,
  type = "HT",
  height = 0,
  filter = TRUE
)
```

Arguments

`sdata`	A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon", "qi". See the data `turtle` for an example. The function filters the input data by a unique "id" (e.g. transmitter number, identifier for each animal). "DateTime" is the GMT date & time of each location in class `POSIXct` or `character` with the following format "2012-06-03 01:33:46". "lat" and "lon" are the latitude and longitude of each location in decimal degrees. "qi" is the quality index associated with each location fix. The input values can be either the number of GPS satellites or Argos Location Classes. Argos Location Classes will be converted to numerical values, where "A", "B", "Z" will be replaced with "-1", "-2", "-3" respectively. The greater number indicates a higher accuracy.
`bathymetry`	A stars object containing bathymetric data in metres. Negative and positive values indicate below and above the water respectively. Geographic coordinate system is WGS84.
`bilinear`	Logical. This defines a method for how to extract cell values from the *bathymetry* layer. Options are bilinear (TRUE) or nearest neighbour (False) as inherited from `st_extract`.
`qi`	An integer specifying a threshold of quality index. *depthfilter* does not filter a location that is associated with a quality index higher than this threshold. Default is 4.

`tide`	A data frame containing columns with the following headers: "tideDT", "reading", "standard.port". "tideDT" is date & time in class `POSIXct` at each observation. "reading" is the observed tidal height in metres. "standard.port" is the identifier of each tidal station.
`tidal.plane`	A data frame containing columns with the following headers: "standard.port", "secondary.port", "lat", "lon", "timeDiff", "datumDiff". "standard.port" is the identifier for a tidal observation station. "secondary.port" is the identifier for a station at which tide is only predicted using tidal records observed at the related standard port. "lat" and "lon" are the latitude and longitude of each secondary port in decimal degrees. "timeDiff" is the time difference between standard port and its associated secondary port. "datumDiff" is the baseline difference in metres if bathymetry and tidal observations/predictions uses different datum (e.g. LAT and MSL).
`type`	The type of water depth considered in the filtering process. "exp" is for the water depth experienced by the animal at the time. This option may be applicable to species that remain in water at all times (e.g. dugongs, dolphins, etc). "HT" is for the water depth at the nearest high tide (default). This option is useful for animals that use inter-tidal zones at high tide and may remain there even after the tide drops (e.g. some sea turtles).
`height`	A numerical value to adjust the water depth an animal is likely to use. Default is 0 m. This parameter is useful if the minimum water depth used by the animal is known. For example, a dugong is unlikely to use water shallower than its body height (e.g. ~0.5 m) so it may be sensible to consider the fix is an error if the estimated water depth is shallower than its body height. A negative value indicates below the water surface. For the dugong example, to remove locations for which the water depth was <0.5 m, it should be specified as; height = -0.5. By supplying the body height to this argument, all the locations recorded shallower than its body will be removed.
`filter`	Default is TRUE. If FALSE, the function does not filter locations but it still returns estimates of the water depth experienced by the animal at each location.

Details

The function examines each location according to the water depth experienced by the animal or the water depth at the nearest high tide. The function looks for the closest match between each fix and tidal observations or predictions in temporal and spatial scales. When *filter* is disabled, the function does not filter locations but returns the estimated water depth of each location with the tide effect considered (bathymetry + tide).

Value

When *filter* option is enabled, this function filters the input data and returns with two additional columns; "depth.exp", "depth.HT". "depth.exp" is the estimated water depth at each location at the time of location fixing. "depth.HT" is the estimated water depth at the nearest high tide at each location.

Note

The input data must not contain temporal or spatial duplicates.

Author(s)

Takahiro Shimada

References

Shimada T, Limpus C, Jones R, Hazel J, Groom R, Hamann M (2016) Sea turtles return home after intentional displacement from coastal foraging areas. *Marine Biology* 163:1-14 doi:10.1007/s0022701527710

Beaman, R.J. (2010) Project 3DGBR: A high-resolution depth model for the Great Barrier Reef and Coral Sea. *Marine and Tropical Sciences Research Facility (MTSRF) Project 2.5i.1a Final Report*, MTSRF, Cairns, Australia, pp. 13 plus Appendix 1.

See Also

`dupfilter`, `ddfilter`

Examples

```
## Not run:

#### Load data sets
## Fastloc GPS data obtained from a green turtle
data(turtle)

## Bathymetry model developed by Beaman (2010)
data(bathymodel)

## A tidal plane for the example site
data(tidalplane)

## Tidal observations and predictions for the example site
data(tidedata)

## Maps for the example site
data(SandyStrait)


#### Remove temporal and/or spatial duplicates and biologically unrealistic fixes
turtle.dd <- ddfilter(dupfilter(turtle))


#### Apply depthfilter
turtle <- depthfilter(sdata = turtle.dd, bathymetry = bathymodel,
tide = tidedata, tidal.plane = tidalplane)


#### Plot data removed or retained by depthfilter
to_map(turtle.dd, bgmap = SandyStrait, point.bg = "red", point.size = 2, line.size = 0.5,
        axes.lab.size = 0, title.size = 0, multiplot = FALSE)[[1]] +
geom_point(aes(x = lon, y = lat), data = turtle, size = 2, fill = "yellow", shape = 21)+
geom_point(aes(x = x, y = y), data = data.frame(x = c(152.68, 152.68), y = c(-25.3, -25.34)),
```

```
           size = 3, fill = c("yellow", "red"), shape = 21) +
annotate("text", x = c(152.7, 152.7), y = c(-25.3, -25.34), label = c("Retained", "Removed"),
         colour = "black", size = 4, hjust = 0)

## End(Not run)
```

`distfilter` *Filter locations by distance*

Description

This function removes locations that are located beyond a specified distance.

Usage

```
distfilter(sdata, max.dist = 100, method = 1, ia = NA)
```

Arguments

`sdata`	A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon". See the data `turtle` for an example. The function filters the input data by a unique "id" (e.g. transmitter number, identifier for each animal). "DateTime" is the GMT date & time of each location in class `POSIXct` or `character` with the following format "2012-06-03 01:33:46". "lat" and "lon" are the latitude and longitude of each location in decimal degrees.
`max.dist`	A numeric value specifying a threshold of distance between successive locations. Default is 100 km.
`method`	An integer specifying how locations should be filtered with *max.dist*. A location is removed if the distance from a previous and(1)/or(2) to a subsequent location exceeds *max.dist*. Default is 1 (both way).
`ia`	An integer (0 to 180) specifying an inner angle (in degrees) between consecutive locations, beyond which the locations are considered potential outliers. Default (NA) ignores this option. See details.

Details

This function removes locations if the distance from a previous and/or to a subsequent location exceeds *max.dist* and the inner angle is less than *ia*. If *ia* is NA (default), inner angles are not considered in the filtering.

Value

The input data is returned without locations identified by this filter. The following columns are added: "pDist", "sDist", 'inAng'. "pDist" and "sDist" are straight distances in kilometres from a previous and to a subsequent fix respectively. "inAng" is the degree between the bearings of lines joining successive location points.

Author(s)

Takahiro Shimada

Examples

```
#### Load data sets
## Fastloc GPS data obtained from a green turtle
data(turtle)

## A Map for the example site
data(Australia)
data(SandyStrait)


#### Filter temporal and/or spatial duplicates
turtle.dup <- dupfilter(turtle, step.time=1/60, step.dist=0.001)


#### distfilter
turtle.dist <- distfilter(turtle.dup, max.dist = 50, ia = 20)


#### Plot data removed or retained by ddfilter
## Entire area
p1 <- to_map(turtle.dup, bgmap=Australia, point.size = 2, line.size = 0.5, axes.lab.size = 0,
            multiplot = FALSE, point.bg = "red",
            title.size=15, title="Entire area")[[1]] +
  geom_point(aes(x=lon, y=lat), data=turtle.dist, size=2, fill="yellow", shape=21)+
  geom_point(aes(x=x, y=y), data=data.frame(x=c(154, 154), y=c(-22, -22.5)),
             size=3, fill=c("yellow", "red"), shape=21) +
  annotate("text", x=c(154.3, 154.3), y=c(-22, -22.5), label=c("Retained", "Removed"),
           colour="black", size=4, hjust = 0)

## Zoomed in
p2 <- to_map(turtle.dup, bgmap=SandyStrait, xlim=c(152.7, 153.2), ylim=(c(-25.75, -25.24)),
            axes.lab.size = 0, point.size = 2, point.bg = "red", line.size = 0.5,
            multiplot = FALSE, title.size=15, title="Zoomed in")[[1]] +
geom_path(aes(x=lon, y=lat), data=turtle.dist, linewidth=0.5, colour="black", linetype=1) +
geom_point(aes(x=lon, y=lat), data=turtle.dist, size=2, colour="black", shape=21, fill="yellow")

gridExtra::marrangeGrob(list(p1, p2), nrow=1, ncol=2)
```

`dupfilter` *Filter temporal and/or spatial duplicates*

Description

Function to filter temporal and spatial duplicates in tracking data and retain only a single fix per time and location.

Usage

```
dupfilter(
  sdata,
  step.time = 0,
  step.dist = 0,
  conditional = FALSE,
  no.cores = 1
)
```

Arguments

`sdata`	A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon", "qi". See the data `turtle` for an example. The function filters the input data by a unique "id" (e.g. transmitter number, identifier for each animal). "DateTime" is the GMT date & time of each location in class `POSIXct` or `character` with the following format "2012-06-03 01:33:46". "lat" and "lon" are the latitude and longitude of each location in decimal degrees. "qi" is the quality index associated with each location fix. The input values can be either the number of GPS satellites or Argos Location Classes. Argos Location Classes will be converted to numerical values, where "A", "B", "Z" will be replaced with "-1", "-2", "-3" respectively. The greater number indicates a higher accuracy.
`step.time`	Consecutive locations less than or equal to *step.time* apart are considered temporal duplicates. Default is 0 hours.
`step.dist`	Consecutive locations less than or equal to *step.dist* apart are considered spatial duplicates. Default is 0 kilometres.
`conditional`	If TRUE, spatial duplicates are filtered only if they are less than or equal to *step.time* apart. Default is FALSE.
`no.cores`	An integer specifying the number of cores used for parallel computing. Alternatively, type in 'detect' to use the maximum number of available cores minus one.

Details

This function filters temporal and spatial duplicates in tracking data. It first filters temporally and spatially exact locations. It then looks for temporal duplicates and retains a fix with the highest quality index. When temporal or spatial duplicates are associated with the same quality index, the function retains a location that is nearest from a previous and to a subsequent location.

Value

The input data frame is returned containing only a single fix (latitude/longitude pair) per time and location. The following columns are added: "pTime", "sTime", "pDist", "sDist". "pTime" and "sTime" are hours from a previous and to a subsequent fix respectively. "pDist" and "sDist" are straight distances in kilometres from a previous and to a subsequent fix respectively.

Author(s)

Takahiro Shimada

References

Shimada T, Limpus C, Jones R, Hazel J, Groom R, Hamann M (2016) Sea turtles return home after intentional displacement from coastal foraging areas. *Marine Biology* 163:1-14 doi:10.1007/s0022701527710

See Also

`dupfilter_exact`, `dupfilter_qi`, `dupfilter_time`, `dupfilter_space`, `track_param`

Examples

```
#### Load data sets
## Fastloc GPS data obtained from a green turtle
data(turtle)


#### Apply dupfilter
turtle.dup <- dupfilter(turtle)
```

`dupfilter_exact` *Filter temporally and spatially exact duplicates*

Description

Function to filter temporally and spatially exact locations in tracking data.

Usage

```
dupfilter_exact(sdata)
```

Arguments

`sdata`	A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon", "qi". See the data `turtle` for an example. The function filters the input data by a unique "id" (e.g. transmitter number, identifier for each animal). "DateTime" is the GMT date & time of each location in class `POSIXct` or `character` with the following format "2012-06-03 01:33:46". "lat" and "lon" are the latitude and longitude of each location in decimal degrees. "qi" is the quality index associated with each location fix. The input values can be either the number of GPS satellites or Argos Location Classes. Argos Location Classes will be converted to numerical values, where "A", "B", "Z" will be replaced with "-1", "-2", "-3" respectively. The greater number indicates a higher accuracy.

Details

This is a partial component of `dupfilter`, although works as a stand-alone function. It looks for temporally and spatially exact locations and retains only a single fix (latitude/longitude pair) per time and location.

Value

The input data frame is returned with temporally and spatially exact duplicates removed.

Author(s)

Takahiro Shimada

References

Shimada T, Limpus C, Jones R, Hazel J, Groom R, Hamann M (2016) Sea turtles return home after intentional displacement from coastal foraging areas. *Marine Biology* 163:1-14 doi:10.1007/s0022701527710

See Also

`dupfilter`, `dupfilter_qi`, `dupfilter_time`, `dupfilter_space`

`dupfilter_qi` *Filter temporal duplicates by quality index*

Description

Function to filter temporal duplicates in tracking data by quality index.

Usage

```
dupfilter_qi(sdata = sdata, step.time = 0)
```

Arguments

`sdata`	A data frame containing columns with the following headers: "id", "DateTime", "qi". See the data `turtle` for an example. The function filters the input data by a unique "id" (e.g. transmitter number, identifier for each animal). "DateTime" is the GMT date & time of each location in class `POSIXct` or `character` with the following format "2012-06-03 01:33:46". "qi" is the quality index associated with each location fix. The input values can be either the number of GPS satellites or Argos Location Classes. Argos Location Classes will be converted to numerical values, where "A", "B", "Z" will be replaced with "-1", "-2", "-3" respectively. The greater number indicates a higher accuracy.
`step.time`	Consecutive locations less than or equal to *step.time* apart are considered temporal duplicates. Default is 0 hours.

Details

This function is a partial component of `dupfilter`, although works as a stand-alone function. It looks for temporal duplicates and retains a fix with the highest quality index.

Value

The input data frame is returned with temporal duplicates removed by the quality index. The following columns are added: "pTime", "sTime". "pTime" and "sTime" are hours from a previous and to a subsequent fix respectively.

Author(s)

Takahiro Shimada

References

Shimada T, Limpus C, Jones R, Hazel J, Groom R, Hamann M (2016) Sea turtles return home after intentional displacement from coastal foraging areas. *Marine Biology* 163:1-14 doi:10.1007/s0022701527710

See Also

`dupfilter`, `dupfilter_exact`, `dupfilter_time`, `dupfilter_space`, `track_param`

`dupfilter_space` *Filter spatial duplicates*

Description

Function to filter spatial duplicates in tracking data.

Usage

```
dupfilter_space(
  sdata,
  step.time = 0,
  step.dist = 0,
  conditional = FALSE,
  no.cores = 1
)
```

Arguments

`sdata`	A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon", "qi". See the data `turtle` for an example. The function filters the input data by a unique "id" (e.g. transmitter number, identifier for each animal). "DateTime" is the GMT date & time of each location in class `POSIXct` or `character` with the following format "2012-06-03 01:33:46". "lat" and "lon" are the latitude and longitude of each location in decimal degrees. "qi" is the quality index associated with each location fix. The input values can be either the number of GPS satellites or Argos Location Classes. Argos Location Classes will be converted to numerical values, where "A", "B", "Z" will be replaced with "-1", "-2", "-3" respectively. The greater number indicates a higher accuracy.

`step.time`	Consecutive locations less than or equal to *step.time* apart are considered temporal duplicates. Default is 0 hours.
`step.dist`	Consecutive locations less than or equal to *step.dist* apart are considered spatial duplicates. Default is 0 kilometres.
`conditional`	If TRUE, spatial duplicates are filtered only if they are less than or equal to *step.time* apart. Default is FALSE.
`no.cores`	An integer specifying the number of cores used for parallel computing. Alternatively, type in 'detect' to use the maximum number of available cores minus one.

Details

This function is a partial component of `dupfilter`, although works as a stand-alone function. First it identifies spatial duplicates by searching for consecutive fixes that were located within *step.dist*. For each group of spatial duplicates, the function then retains a single fix that is nearest from a previous and to a subsequent location.

Value

The input data frame is returned with spatial duplicates removed. The following columns are added: "pTime", "sTime", "pDist", "sDist". "pTime" and "sTime" are hours from a previous and to a subsequent fix respectively. "pDist" and "sDist" are straight distances in kilometres from a previous and to a subsequent fix respectively.

Note

A minimum of two locations per id is required.

Author(s)

Takahiro Shimada

References

Shimada T, Limpus C, Jones R, Hazel J, Groom R, Hamann M (2016) Sea turtles return home after intentional displacement from coastal foraging areas. *Marine Biology* 163:1-14 doi:10.1007/s0022701527710

See Also

`dupfilter`, `dupfilter_exact`, `dupfilter_time`, `dupfilter_qi`, `track_param`

`dupfilter_time` *Filter temporal duplicates*

Description

Function to filter temporal duplicates that are associated with the same quality index.

Usage

```
dupfilter_time(sdata, step.time = 0, no.cores = 1)
```

Arguments

`sdata`	A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon", "qi". See the data `turtle` for an example. The function filters the input data by a unique "id" (e.g. transmitter number, identifier for each animal). "DateTime" is the GMT date & time of each location in class `POSIXct` or `character` with the following format "2012-06-03 01:33:46". "lat" and "lon" are the latitude and longitude of each location in decimal degrees. "qi" is the quality index associated with each location fix. The input values can be either the number of GPS satellites or Argos Location Classes. Argos Location Classes will be converted to numerical values, where "A", "B", "Z" will be replaced with "-1", "-2", "-3" respectively. The greater number indicates a higher accuracy.
`step.time`	Consecutive locations less than or equal to *step.time* apart are considered temporal duplicates. Default is 0 hours.
`no.cores`	An integer specifying the number of cores used for parallel computing. Alternatively, type in 'detect' to use the maximum number of available cores minus one.

Details

This is a partial component of `dupfilter`, although works as a stand-alone function. First it identifies temporal duplicates by searching for consecutive locations that were obtained within *step.time*. For each group of temporal duplicates, the function then retains a single fix that is nearest from a previous and to a subsequent location.

Value

The input data frame is returned with temporal duplicates removed. The following columns are added: "pTime", "sTime", "pDist", "sDist". "pTime" and "sTime" are hours from a previous and to a subsequent fix respectively. "pDist" and "sDist" are straight distances in kilometres from a previous and to a subsequent fix respectively.

Author(s)

Takahiro Shimada

References

Shimada T, Limpus C, Jones R, Hazel J, Groom R, Hamann M (2016) Sea turtles return home after intentional displacement from coastal foraging areas. *Marine Biology* 163:1-14 doi:10.1007/s0022701527710

See Also

`dupfilter`, `dupfilter_exact`, `dupfilter_qi`, `dupfilter_space`, `track_param`

`flatback` *Flatback turtle tracking data*

Description

Satellite tracking data of 15 flatback turtles (*Natator depressus*) that nested in Curtis Island, Australia. This sample data is a subset of the tracking data used in Shimada et al. (2021).

Usage

```
flatback
```

Format

A data frame with 1020 rows and 4 variables:

id identifier for each animal.

DateTime GMT date & time of each location in class `POSIXct`.

x longitude in UTM.

y latitude in UTM.

Source

Shimada T, Thums M, Hamann M, Limpus CJ, Hays GC, FitzSimmons N, Wildermann NE, Duarte CD, Meekan MG (2021) Optimising sample sizes for animal distribution analysis using tracking data. *Methods in Ecology and Evolution* 12(2):288-297 doi:10.1111/2041210X.13506

`percent_vol` *UD percent volume*

Description

Function to calculate a percent volume on a utilisation distribution (UD)

Usage

```
percent_vol(x, percent = 100)
```

Arguments

`x`	A vector containing the probability density.
`percent`	An integer specifying the percent volume of a UD to be considered.

Details

This function calculates a percent volume on a UD. The probability beyond the specified range will be assigned with a zero value.

Value

A vector containing the specified percent volume.

Author(s)

Takahiro Shimada

`SandyStrait` *A map of Sandy Strait, Australia*

Description

This map layer outlines the coast around Sandy Strait, Australia.

Usage

```
SandyStrait
```

Format

A data.frame

`tidalplane` *Tidal plane table for Sandy Strait, Australia*

Description

A semidiurnal tidal plane table containing the height of the mean tidal planes and the average time differences of tide at different locations within Sandy Strait.

Usage

```
tidalplane
```

Format

A data frame with 2 rows and 6 variables:

standard.port identifier for a tidal observation station.

secondary.port identifier for a station at which tide is only predicted using the tidal records observed at the related standard port.

lat latitude in decimal degrees.

lon longitude in decimal degrees.

timeDiff time difference between standard port and its associated secondary port.

datumDiff baseline difference in metres between the bathymetry model and tidal observations/predictions, if each data uses different datum (e.g. LAT and MSL).

Source

The State of Queensland (Department of Transport and Main Roads), Tidal planes.

`tidedata` *Tidal data for Sandy Strait, Australia*

Description

A dataset containing tidal observations recorded at Bundaberg, Australia

Usage

```
tidedata
```

Format

A data frame with 26351 rows and 3 variables:

tideDT GMT date & time of each observation in class `POSIXct`.

reading observed tidal height in metres.

standard.port identifier of the tidal station.

Source

The State of Queensland (Department of Transport and Main Roads), Tidal data.

`to_map` *Plot location data on a map*

Description

Function to plot tracking data on a map or a satellite image.

Usage

```
to_map(
  sdata,
  xlim = NULL,
  ylim = NULL,
  margin = 10,
  bgmap = NULL,
  google.key = NULL,
  map.bg = "grey",
  map.col = "black",
  zoom = NULL,
  point.bg = "yellow",
  point.col = "black",
  point.symbol = 21,
  point.size = 1,
  line.col = "lightgrey",
  line.type = 1,
  line.size = 0.5,
  title = "id",
  title.size = 11,
  axes.text.size = 11,
  axes.lab.size = 11,
  multiplot = TRUE,
  nrow = 1,
  ncol = 1
)
```

Arguments

`sdata`	A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon". The function creates a map for each unique "id" (e.g. transmitter number, identifier for each animal). "DateTime" is the GMT date & time of each location in class `POSIXct` or `character` with the following format "2012-06-03 01:33:46". "lat" and "lon" are the latitude and longitude of each location in decimal degrees.

`xlim`, `ylim`	Limits for x and y axes. If not specified, the values are determined as the maximum range of the input data plus an additional margin (see *margin*).
`margin`	Set the amount of spaces added around the periphery of the plot. The value is scaled to the plot. The smaller value increases the margin.
`bgmap`	A data frame of a background map data, containing the following headers: "long", "lat", "group". If not specified, the `world` map is used. Google Maps ("terrain", "satellite", "roadmap", "hybrid") can also be queried.
`google.key`	If Google Maps are queried, a valid API key (a string) needs to be specified here. See `register_google` for details.
`map.bg`	Background colour of the map. This argument is ignored when any Google Maps is selected.
`map.col`	Outline colour of the map. This argument is ignored when any Google Maps is selected.
`zoom`	Map zoom for Google Maps. Default (NULL) to estimate the zoom from each data set. For other options, see `get_map` for details.
`point.bg`	The colour to fill in a symbol.
`point.col`	The colour for the outline of a symbol.
`point.symbol`	An integer or a string to specify the symbol type. See `shape` for details.
`point.size`	An integer to specify the size of the symbol.
`line.col`	The colour of the line that connects consecutive points.
`line.type`	The type of the line that connects consecutive points. See `linetype` for details.
`line.size`	An integer to specify the thickness (width) of the line that connects consecutive points.
`title`	The main title for each plot. If not specified, the "id" will be used.
`title.size`	An integer to specify the size of the title.
`axes.text.size`	An integer to specify the size of the axes characters.
`axes.lab.size`	An integer to specify the size of the axes labels.
`multiplot`	Logical. If TRUE (default), multiple plots are displayed on the same page.
`nrow`	An integer to specify the number of rows in the multiple plot page.
`ncol`	An integer to specify the number of columns in the multiple plot page.

Value

An arrangelist is returned when *multiplot* is TRUE. Otherwise a list is returned.

Author(s)

Takahiro Shimada

See Also

`dupfilter`, `ddfilter`, `vmax`, `vmaxlp`

Examples

```
#### Load data sets
## Fastloc GPS data obtained from two green turtles
data(turtle)
data(turtle2)
turtles<-rbind(turtle, turtle2)

#### Filter temporal and/or spatial duplicates
turtle.dup <- dupfilter(turtles, step.time=5/60, step.dist=0.001)


#### ddfilter
V <- vmax(turtle.dup)
VLP <- vmaxlp(turtle.dup)
turtle.dd <- ddfilter(turtle.dup, vmax=V, vmaxlp=VLP)


#### Plot filtered data for each animal
## using the low-resolution world map
to_map(turtle.dd, point.size = 2, line.size = 0.5, axes.lab.size = 0, ncol=2, nrow=1)

## Not run:
## using the high-resolution google satellite images
to_map(turtle.dd, bgmap = "satellite", google.key = "key", ncol=2)

## End(Not run)
```

`track_param` *Calculate parameters between locations*

Description

Calculate time, distance, speed, and inner angle between successive locations

Usage

```
track_param(
  sdata,
  param = c("time", "distance", "speed", "angle", "mean speed", "mean angle"),
  days = 2
)
```

Arguments

`sdata`	A data.frame or a list of data.frames containing columns with the following headers: "id", "DateTime", "lat", "lon". The function calculates each movement parameter by a unique "id" (e.g. transmitter number, identifier for each animal) if the input is a data.frame, or by each element of the list if the input is a list. "DateTime" is the GMT date & time of each location in class `POSIXct` or

	`character` with the following format "2012-06-03 01:33:46". "lat" and "lon" are the latitude and longitude of each location in decimal degrees.
`param`	A string or vector specifying movement parameters to be calculated. Options are 'time', 'distance', 'speed', 'angle', 'mean speed' and 'mean angle'. See *details*.
`days`	A numeric value specifying the number of days to calculate mean speeds and angles. This argument is only used when 'mean speed' and/or 'mean angle' are selected in *param*.

Details

This function calculates various parameters of tracks. time (h), distance (km), speed (km/h) and inner angle (degrees) are calculated from each pair of successive locations. mean speed (km/h) and angle (degrees) are calculated from locations over a specified number of days.

Value

The input data is returned with new columns containing the requested parameters. "pTime" and "sTime" are hours from a previous and to a subsequent fix respectively. "pDist" and "sDist" are straight distances in kilometres from a previous and to a subsequent fix respectively. "pSpeed" and "sSpeed" are linear speed (km/h) from a previous and to a subsequent fix respectively. "inAng" is the degree between the bearings of lines joining successive location points. "meanSpeed" and "meanAngle" are the mean speed and degree over a specified number of days.

Author(s)

Takahiro Shimada

Examples

```
#### Load turtle tracking data
data(turtle)


#### Filter temporal and/or spatial duplicates
turtle.dup <- dupfilter(turtle, step.time=5/60, step.dist=0.001)


#### ddfilter
turtle.dd <- ddfilter(turtle.dup, vmax=9.9, qi=4, ia=90, vmaxlp=2.0)


#### Mean speed over 2 days
mean.speed <- track_param(turtle.dd, param = c('speed', 'mean speed'), days=2)


#### Plot data
ggplot(data = mean.speed, aes(x=lon, y=lat)) +
geom_path(colour = 'grey') +
geom_point(aes(colour=meanSpeed))
```

`turtle` *Green turtle tracking data*

Description

A dataset containing Fastloc GPS locations of a green turtle tracked in Sandy Strait, Australia.

Usage

```
turtle
```

Format

A data frame with 429 rows and 5 variables:

id identifier for each animal.

DateTime GMT date & time of each location in class `POSIXct`.

lat latitude in decimal degrees.

lon longitude in decimal degrees.

qi quality index associated with each location fix. The input values can be either the number of GPS satellites or Argos Location Classes. Argos Location Classes will be converted to numerical values, where "A", "B", "Z" will be replaced with "-1", "-2", "-3" respectively. The greater number indicates a higher accuracy.

Source

Shimada T, Jones R, Limpus C, Groom R, Hamann M (2016) Long-term and seasonal patterns of sea turtle home ranges in warm coastal foraging habitats: Implications for conservation. *Marine Ecology Progress Series* 562:163-179. doi:10.3354/meps11972

`turtle2` *Green turtle tracking data 2*

Description

A dataset containing Fastloc GPS locations of a green turtle tracked in Moreton Bay, Australia.

Usage

```
turtle2
```

Format

A data frame with 276 rows and 5 variables:

id identifier for each animal.

DateTime GMT date & time of each location in class `POSIXct`.

lat latitude in decimal degrees.

lon longitude in decimal degrees.

qi quality index associated with each location fix. The input values can be either the number of GPS satellites or Argos Location Classes. Argos Location Classes will be converted to numerical values, where "A", "B", "Z" will be replaced with "-1", "-2", "-3" respectively. The greater number indicates a higher accuracy.

Source

Shimada T, Jones R, Limpus C, Groom R, Hamann M (2016) Long-term and seasonal patterns of sea turtle home ranges in warm coastal foraging habitats: Implications for conservation. *Marine Ecology Progress Series* 562:163-179. doi:10.3354/meps11972

`ud_matrix` *A matrix containing probability distributions of flatback turtles*

Description

Inter-nesting utilisation distributions of 15 flatback turtles (*Natator depressus*) that nested in Curtis Island, Australia. The UDs were calculated using the sample tracking data `flatback` and reduced grid resolution (1 km) instead of 50m as used in Shimada et al. (2021). See GitHub for an example code of UD estimation.

Usage

```
ud_matrix
```

Format

A matrix

Source

Shimada T, Thums M, Hamann M, Limpus CJ, Hays GC, FitzSimmons N, Wildermann NE, Duarte CD, Meekan MG (2021) Optimising sample sizes for animal distribution analysis using tracking data. *Methods in Ecology and Evolution* 12(2):288-297 doi:10.1111/2041210X.13506

`ud_raster` *A list of raster data containing probability distributions of flatback turtles*

Description

Inter-nesting utilisation distributions of 15 flatback turtles (*Natator depressus*) that nested in Curtis Island, Australia. The UDs were calculated using the sample tracking data `flatback` and reduced grid resolution (1 km) instead of 50m as used in Shimada et al. (2021). See GitHub for an example code of UD estimation.

Usage

```
ud_raster
```

Format

A list of 15 stars objects

Source

Shimada T, Thums M, Hamann M, Limpus CJ, Hays GC, FitzSimmons N, Wildermann NE, Duarte CD, Meekan MG (2021) Optimising sample sizes for animal distribution analysis using tracking data. *Methods in Ecology and Evolution* 12(2):288-297 doi:10.1111/2041210X.13506

`vmax` *Maximum linear speed*

Description

Function to estimate the maximum linear speed between two consecutive locations.

Usage

```
vmax(sdata, qi = 5, method = "ML", prob = 0.99, ...)
```

Arguments

`sdata`	A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon", "qi". See the data `turtle` for an example. The function filters the input data by a unique "id" (e.g. transmitter number, identifier for each animal). "DateTime" is the GMT date & time of each location in class `POSIXct` or `character` with the following format "2012-06-03 01:33:46". "lat" and "lon" are the latitude and longitude of each location in decimal degrees. "qi" is the quality index associated with each location fix. The input values can be either the number of GPS satellites or Argos Location Classes. Argos Location Classes will be converted to numerical values, where "A", "B", "Z" will be replaced with "-1", "-2", "-3" respectively. The greater number indicates a higher accuracy.

`qi`	An integer specifying the lowest quality index of a location that is qualified to be used in the estimation. Default is 5 (e.g. 5 GPS satellite or more).
`method`	Available options are "sample" (i.e. sample quantile - see `quantile`) and "ML" (maximum likelihood estimation). Default is "ML". See details.
`prob`	A value (0 to 1) specifying the sample quantile or cumulative probability for linear speed. Values beyond this threshold are considered 'outliers' and excluded from estimation of maximum linear speed. Default is 0.99. See details.
`...`	Extra arguments passed to `dupfilter`.

Details

The function first calculates the linear speed between each pair of two consecutive locations. Some of the calculated linear speed can be inaccurate when the input data contains inaccurate locations (e.g. outliers). The function can discard the implausible outliers by excluding extreme values using either the "sample" or "ML" method. The "sample" method simply discards values that lie beyond the specified quantile. If the "ML" method is selected, it is assumed that the linear speed follow a Gamma distribution. The distribution parameters are derived via maximum likelihood estimation using the `optim` function. The linear speed at the given quantile or cumulative probability (e.g. 0.99) represents the maximum linear speed at which an animal would travel between two consecutive locations.

Value

Maximum linear speed (vmax) estimated from the input data. The unit is km/h.

Author(s)

Takahiro Shimada

References

Shimada T, Jones R, Limpus C, Hamann M (2012) Improving data retention and home range estimates by data-driven screening. *Marine Ecology Progress Series* 457:171-180 doi:10.3354/meps09747

See Also

`ddfilter`, `ddfilter_speed`, `track_param`, `dupfilter`

`vmaxlp` *Maximum one-way linear speed of a loop trip*

Description

Function to estimate the maximum one-way linear speed of a loop trip.

Usage

```
vmaxlp(sdata, qi = 4, nloc = 5, method = "ML", prob = 0.99, ...)
```

Arguments

`sdata`	A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon", "qi". See the data `turtle` for an example. The function filters the input data by a unique "id" (e.g. transmitter number, identifier for each animal). "DateTime" is the GMT date & time of each location in class `POSIXct` or `character` with the following format "2012-06-03 01:33:46". "lat" and "lon" are the recorded latitude and longitude in decimal degrees. "qi" is the quality index associated with each location fix. The input values can be either the number of GPS satellites or Argos Location Classes. Argos Location Classes will be converted to numerical values, where "A", "B", "Z" will be replaced with "-1", "-2", "-3" respectively. The greater number indicates a higher accuracy.
`qi`	An integer specifying the minimum quality index associated with a location used for the estimation. Default is 4 (e.g. 4 GPS satellite or more).
`nloc`	An integer specifying the minimum number of successive locations to be considered a loop trip.
`method`	Available options are "sample" (i.e. sample quantile - see `quantile`) and "ML" (maximum likelihood estimation - see details). Default is "ML".
`prob`	A value (0 to 1) specifying the sample quantile or cumulative probability for one-way linear speed of a loop trip. Values beyond this threshold are considered ’outliers’ and excluded from estimation of maximum one-way linear speed of a loop trip. Default is 0.99. See details.
`...`	Extra arguments passed to `dupfilter`.

Details

The function first detects a "loop trip". Loop trip behaviour is represented by spatial departure and return involving more than 3 consecutive locations (Shimada et al. 2012). The function calculates the net (i.e. straight-line) distance between the departure and turning point as well as the turning point and return location of a loop trip. It then calculates the one-way travelling speed to or from each turning point for each loop trip. To exclude implausible outliers, the function discards extreme values based on the specified quantile or an estimated probability distribution for the loop trip speed, depending on the selected *method*. If the "ML" method is selected, a Gamma distribution is assumed and the shape and scale parameters are estimated via maximum likelihood estimation using the `optim` function. The maximum value within a given quantile or probability range (e.g. 0.99) represents the maximum one-way linear speed at which an animal would travel during a loop trip.

Value

Maximum one-way linear speed of a loop trip (vmaxlp) estimated from the input data. The unit km/h.

Note

The input data must not contain temporal or spatial duplicates. A minimum of 8 locations are required.

Author(s)

Takahiro Shimada

References

Shimada T, Jones R, Limpus C, Hamann M (2012) Improving data retention and home range estimates by data-driven screening. *Marine Ecology Progress Series* 457:171-180 doi:10.3354/meps09747

See Also

`ddfilter`, `ddfilter_loop`, `track_param`, `dupfilter`

Index