Package ‘SDLfilter’

February 27, 2019

Type Package
Title Filtering Satellite-Derived Locations
Version 1.2.1
Date 2019-02-19
Author Takahiro Shimada
Maintainer Takahiro Shimada <taka.shimada@gmail.com>
Description Functions to filter GPS and/or Argos locations. The provided filters remove temporal and spatial duplicates, fixes located at a given height from estimated high tide line, and locations with high error as proposed in Shimada et al. (2012) <doi:10.3354/meps09747> and Shimada et al. (2016) <doi:10.1007/s00227-015-2771-0>.
Depends R (>= 2.10), ggplot2
Imports sp, raster, trip, data.table, geosphere, gmap, gridExtra, ggsn, stats, maps
License GPL-2 | file LICENSE
URL https://github.com/TakahiroShimada/SDLfilter
BugReports https://github.com/TakahiroShimada/SDLfilter/issues
LazyData true
Encoding UTF-8
RoxygenNote 6.1.1
NeedsCompilation no
Repository CRAN
Date/Publication 2019-02-27 00:00:07 UTC

R topics documented:

Australia ................................................................. 2
bathymodel ............................................................. 2
dfdfilter ................................................................. 3
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
ddfilter

Format
A RasterLayer

Source
https://www.deepeef.org/

References

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, maxvlp = 1.8, qi = 4, ia = 90, method = 2)

Arguments
- **sdata**: A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon", "qi". This filter is independently applied to a subset of data grouped by the unique "id". "DateTime" is date & time in class POSIXct. "lat" and "lon" are the recorded latitude and longitude in decimal degrees. "qi" is the numerical quality index associated with each fix where the greater number represents better quality (e.g. number of GPS satellites used for estimation).
- **vmax**: A numeric vector specifying threshold speed both from a previous and to a subsequent fix. Default is 8.9km/h. If this value is unknown, the function "est.vmax" can be used to estimate the value based on the supplied data.
- **maxvlp**: A numeric value specifying threshold speed during a loop trip. Default is 1.8 km/h. If this value is unknown, the function "est.maxvlp" can be used to estimate the value based on the supplied data.
- **qi**: An integer specifying threshold quality index during a loop trip. Default is 4.
- **ia**: An integer specifying threshold inner angle during a loop trip. Default is 90 degrees.
- **method**: An integer specifying how locations are filtered by speed. 1 = a location is removed if the speed EITHER from a previous and to a subsequent location exceeds a given threshold speed. 2 = a location is removed if the speed BOTH from a previous and to a subsequent location exceeds a given threshold speed. Default is 2.
Details

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

Value

A data frame 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 angle between the bearings of lines joining successive location points.

Author(s)

Takahiro Shimada

References


See Also

ddfilter.speed, ddfilter.loop, est.vmax, est.maxvlp

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 <- est.vmax(turtle.dup)
VLP <- est.maxvlp(turtle.dup)
turtle.dd <- ddfilter(turtle.dup, vmax=V, maxvlp=VLP)

## Or using user specified threshold speeds
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

```r
ddfilter.loop(sdata, qi = 4, ia = 90, maxvlp = 1.8)
```

Arguments

- `sdata`: A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon", "qi". This filter is independently applied to a subset of data grouped by the unique "id". "DateTime" is date & time in class `POSIXct`. "lat" and "lon" are the recorded latitude and longitude in decimal degrees. "qi" is the numerical quality index associated with each fix where the greater number represents better quality (e.g. number of GPS satellites used for estimation).
- `qi`: An integer specifying threshold quality index during a loop trip. Default is 4 satellites.
- `ia`: An integer specifying threshold inner angle during a loop trip. Default is 90 degrees.
- `maxvlp`: A numeric value specifying threshold speed during a loop trip. Default is 1.8 km/h. If this value is unknown, the function "est.maxvlp" can be used to estimate the value based on the supplied data.
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 "maxvlp". If "maxvlp" is unknown, it can be estimated using the function "est.maxvlp".

Value
A data frame 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 angle between the bearings of lines joining successive location points.

Author(s)
Takahiro Shimada

References

See Also
ddfilter, ddfilter.speed, est.maxvlp

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 = 2)

Arguments
sdata A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon", "qi". This filter is independently applied to a subset of data grouped by the unique "id". "DateTime" is date & time in class POSIXct. "lat" and "lon" are the recorded latitude and longitude in decimal degrees. "qi" is the numerical quality index associated with each fix where the greater number represents better quality (e.g. number of GPS satellites used for estimation).
**depthfilter**

A numeric value specifying threshold speed both from a previous and to a subsequent fix. Default is 8.9 km/h. If this value is unknown, the function "est.vmax" can be used to estimate the value based on the supplied data.

**method**

An integer specifying how locations are filtered by speed. 1 = a location is removed if the speed EITHER from a previous and to a subsequent location exceeds a given threshold speed. 2 = a location is removed if the speed BOTH from a previous and to a subsequent location exceeds a given threshold speed. Default is FALSE.

**Details**

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

**Value**

A data frame 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**


**See Also**

*ddfilter, ddfilter.loop, est.vmax*

---

**depthfilter**

*Filter locations by water depth*

**Description**

Function to remove fixes located at a given height from the high tide line.

**Usage**

```r
depthfilter(sdata, bathymetry, extract = "bilinear", tide, qi = 4,
depth = 0, tidal.plane, filter = TRUE)
```
Arguments

sdata  A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon", "qi". This filter is independently applied to a subset of data grouped by the unique "id", "DateTime" is date & time in class POSIXct. "lat" and "lon" are the recorded latitude and longitude in decimal degrees. "qi" is the numerical quality index associated with each fix where the greater number represents better quality (e.g. number of GPS satellites used for estimation).

bathymetry  object of class "RasterLayer" containing bathymetric data in meters. Geographic coordinate system is WGS84.

extract  Method to extract cell values from raster layer inherited from extract function of the raster package. Default is "bilinear". See extract for details.

tide  A data frame containing columns with the following headers: "tideDT", "reading", "standard.port". "tideDT" is date & time in class POSIXct for each observed tidal height. "reading" is the observed tidal height in meters. "standard.port" is the identifier of each tidal station.

qi  An integer specifying threshold quality index. Fixes associated to a quality index higher than the threshold are excluded from the depthfilter. Default is 4

depth  An integer denoting vertical difference from a high tide line in meters. A positive value indicates above the high tide and a negative value indicates below the high tide. The function removes fixes above the given threshold. Default is 0 m (i.e. high tide line).

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 meters between bathymetry and tidal observations/predictions if each data uses different datum (e.g. LAT and MSL).

filter  Default is TRUE. If FALSE, the function does not filter locations but the depth estimates are returned.

Details

This function removes fixes located at a given height from estimated high tide line when the "filter" option is enabled. The function chooses the closest match between each fix and tidal observations or predictions in temporal and spatial scales in order to estimate height of high tide at the time and location of each fix. It does not filter data when the "filter" option is disabled but it returns the estimated water depth of each location with the tide effect accounted for (bathymetry + tide). The estimated water depths are returned in the "depth.exp" column.

Value

Input data is returned with two columns added; "depth.exp", "depth.HT". "depth.exp" is the estimated water depth at the time of location fixing. "depth.HT" is the estimated water depth at the
high tide nearest to the time and location of each fix. When the "filter" option is enabled, the fixes identified by this filter are removed from the input data.

Note
Input data must not contain temporal or spatial duplicates.

Author(s)
Takahiro Shimada

References

See Also
dupfilter, ddfilter

Examples

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

#### Remove biologically unrealistic fixes
turtle.dd <- ddfilter(turtle.dup, vmax=9.9, qi=4, ia=90, maxvlp=2.0)
```
distance_filter

Filter locations by distance

Description
This function removes locations that are located beyond a given threshold distance.

Usage
distance_filter(sdata, distance1 = 100, distance2 = 100, method = 2)

Arguments

- **sdata**: A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon", "qi". This filter is independently applied to a subset of data grouped by the unique "id". "DateTime" is date & time in class `POSIXct`. "lat" and "lon" are the recorded latitude and longitude in decimal degrees. "qi" is the numerical quality index associated with each fix where the greater number represents better quality (e.g. number of GPS satellites used for estimation).

- **distance1**: A numeric value specifying threshold distance from a previous fix. Default is 100 km.

- **distance2**: A numeric value specifying threshold distance to a subsequent fix. Default is 100 km.

- **method**: An integer specifying how locations are filtered by distance. 1 = a location is removed if the distance EITHER from a previous and to a subsequent location exceeds a given threshold speed. 2 = a location is removed if the distance BOTH from a previous and to a subsequent location exceeds a given threshold speed. Default is FALSE.
Details

This function removes locations if the distance both/either from a previous and to a subsequent location exceeds a given threshold distance.

Value

A data frame is returned without locations identified by this filter. The following columns are added: "pTime", "pDist". "pTime" is hours from a previous fix. "pDist" is straight distance in kilometres from a previous fix.

Author(s)

Takahiro Shimada

dupfilter  Filter temporal and/or spatial duplicates

Description

Function to remove temporal and/or spatial duplicates.

Usage

dupfilter(sdata, step.time = 0, step.dist = 0, conditional = FALSE)

Arguments

sdata  A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon", "qi". This filter is independently applied to a subset of data grouped by the unique "id". "DateTime" is date & time in class POSIXct. "lat" and "lon" are the recorded latitude and longitude in decimal degrees. "qi" is the numerical quality index associated with each fix where the greater number represents better quality (e.g. number of GPS satellites used for estimation).

step.time  A numeric value specifying temporal interval between two consecutive locations. Default is 0 hours. Locations are considered temporal duplicates if the temporal interval is less than or equal to the user specified value.

step.dist  A numeric value specifying spatial interval between two consecutive locations. Default is 0 kilometres. Locations are considered spatial duplicates if the spatial interval is less than or equal to the user specified value.

conditional  If TRUE, spatial duplicates are removed only if temporal interval between the locations is less than the time specified in "step.time". Default is FALSE.

Details

A fix associated with a higher quality index is retained over other duplicated fixes. If temporal duplicates are associated with same quality index, a fix located closest to a previous and a subsequent location is retained.
Value

Input data frame is returned with spatial and 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


See Also

dupfilter.exact, dupfilter.qi, dupfilter.time, dupfilter.space

Examples

#### load data sets
## fastloc gps data obtained from a green turtle
data(turtle)

#### apply dupfilter
turtle.dup <- dupfilter(turtle)

dupfilter.exact(sdata)

Description

A partial component of dupfilter, although works as a stand-alone function. This function removes temporally and spatially exact duplicates.

Usage

dupfilter.exact(sdata)
**Arguments**

sdata A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon", "qi". This filter is independently applied to a subset of data grouped by the unique "id". "DateTime" is date & time in class POSIXct. "lat" and "lon" are the recorded latitude and longitude in decimal degrees. "qi" is the numerical quality index associated with each fix where the greater number represents better quality (e.g. number of GPS satellites used for estimation).

**Details**

This function selects a fix from multiple fixes, which were simultaneously obtained at the same geographical coordinate.

**Value**

Input data frame is returned with temporally and spatially exact 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**


**See Also**

dupfilter, dupfilter.qi, dupfilter.time, dupfilter.space

dupfilter.qi Filter temporal duplicates by quality index

**Description**

A partial component of dupfilter, although works as a stand-alone function. This function removes temporal duplicates according to the associated quality index.

**Usage**

dupfilter.qi(sdata, step.time = 0)
Arguments

sdata  A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon", "qi". This filter is independently applied to a subset of data grouped by the unique "id". "DateTime" is date & time in class POSIXct. "lat" and "lon" are the recorded latitude and longitude in decimal degrees. "qi" is the numerical quality index associated with each fix where the greater number represents better quality (e.g. number of GPS satellites used for estimation).

step.time  A numeric value specifying temporal interval between two consecutive locations. Default is 0 hours. Locations are considered temporal duplicates if the temporal interval is less than or equal to the user specified value.

detials

This function selects a fix from multiple fixes, which were simultaneously obtained but associated with a different level of quality index. The fix with the higher quality index is selected over the others.

Value

Input data frame is returned with temporal duplicates removed according to 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


See Also

dupfilter, dupfilter.exact, dupfilter.time, dupfilter.space

dupfilter.space  Filter spatial duplicates
Usage

dupfilter.space(sdata, step.time = 0, step.dist = 0, conditional = FALSE)

Arguments

sdata        A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon", "qi". This filter is independently applied to a subset of data grouped by the unique "id". "DateTime" is date & time in class POSIXct. "lat" and "lon" are the recorded latitude and longitude in decimal degrees. "qi" is the numerical quality index associated with each fix where the greater number represents better quality (e.g. number of GPS satellites used for estimation).

step.time    A numeric value specifying temporal interval between two consecutive locations. Default is 0 hours. Locations are considered temporal duplicates if the temporal interval is less than or equal to the user specified value.

step.dist    A numeric value specifying spatial interval between two consecutive locations. Default is 0 kilometres. Locations are considered spatial duplicates if the spatial interval is less than or equal to the user specified value.

conditional If TRUE, spatial duplicates are removed only if the temporal interval between the locations is less than the time specified in "step.time". Default is FALSE.

Details

This function selects a fix from multiple fixes which were obtained at the same geographical coordinate. A minimum of two locations per id is required to run this function.

Value

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.

Author(s)

Takahiro Shimada

References


See Also

dupfilter, dupfilter.exact, dupfilter.time, dupfilter.qi
### dupfilter.time

**Filter temporal duplicates**

**Description**

A partial component of dupfilter, although works as a stand-alone function. This function removes temporal duplicates.

**Usage**

```r
dupfilter.time(sdata, step.time = 0)
```

**Arguments**

- **sdata**
  - A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon", "qi". This filter is independently applied to a subset of data grouped by the unique "id". "DateTime" is date & time in class `POSIXct`. "lat" and "lon" are the recorded latitude and longitude in decimal degrees. "qi" is the numerical quality index associated with each fix where the greater number represents better quality (e.g. number of GPS satellites used for estimation).

- **step.time**
  - A numeric value specifying temporal interval between two consecutive locations. Default is 0 hours. Locations are considered temporal duplicates if the temporal interval is less than or equal to the user specified value.

**Details**

This function removes temporal duplicates according to the total distance from a previous and to a subsequent location. A fix with a shorter sum distance is retained.

**Value**

Input data frame is returned with temporal duplicates removed according to total distance from a previous and to a subsequent location. 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**


**See Also**

- `dupfilter`
- `dupfilter.exact`
- `dupfilter.qi`
- `dupfilter.space`
est.maxvlp  

Estimate maximum one-way linear speed of a loop trip

Description

This function estimates the maximum one-way linear speed of a loop trip as described in Shimada et al. (2012).

Usage

est.maxvlp(sdata, qi = 4, prob = 0.99)

Arguments

sdata  A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon", "qi". This filter is independently applied to a subset of data grouped by the unique "id". "DateTime" is date & time in class POSIXct. "lat" and "lon" are the recorded latitude and longitude in decimal degrees. "qi" is the numerical quality index associated with each fix where the greater number represents better quality (e.g. number of GPS satellites used for estimation).

qi  An integer specifying the minimum quality index associated with a location used for the estimation. Default is 4.

prob  A numeric value specifying a probability to obtain sample quantiles. Default is 0.99.

Details

The function first detects a "loop trip". Loop trip behaviour is represented by spatial departure and return with more than 3 consecutive locations (Shimada et al 2012). It then calculates the net (i.e. straight-line) distance between the departure and turning location as well as the turning and return location of the loop trip, and from that calculated the net speed in and out. It discards extreme values based on the quantile specified by a user (default is 0.99). This is to exclude outliers potentially contained in the original data set. The maximum value in the retained dataset (i.e. without outliers) represents the maximum one-way linear speed at which an animal would travel during a loop trip. A minimum of 8 locations are required to run this function.

Value

A vector is returned. The unit is in kilometres per hour.

Note

Input data must not contain temporal or spatial duplicates.

Author(s)

Takahiro Shimada
References


See Also

ddfiter, ddfilter.loop

---

est.vmax

Estimate maximum linear speed

Description

This function estimates the maximum linear speed between two consecutive locations as described in Shimada et al. (2012)

Usage

est.vmax(sdata, qi = 5, prob = 0.99)

Arguments

- **sdata**: A data frame containing columns with the following headers: "id", "DateTime", "lat", "lon", "qi". This filter is independently applied to a subset of data grouped by the unique "id". "DateTime" is date & time in class POSIXct. "lat" and "lon" are the recorded latitude and longitude in decimal degrees. "qi" is the numerical quality index associated with each fix where the greater number represents better quality (e.g. number of GPS satellites used for estimation).

- **qi**: An integer specifying minimum quality index associated with a location used for the estimation. Default is 5.

- **prob**: numeric value of a probability to obtain sample quantiles. Default is 0.99.

Details

The function first calculates the linear speeds between each pair of two consecutive locations. It discards extreme values based on the quantile specified by a user (default is 0.99). This is to deal with outliers potentially contained in the original data set. The maximum value in the retained dataset (i.e. without outliers) represents the maximum linear speed at which an animal would travel between two consecutive locations.

Value

A vector is returned. The unit is in kilometres per hour.

Note

Input data must not contain temporal or spatial duplicates.
plotMap

**Author(s)**

Takahiro Shimada

**References**


**See Also**

ddfilter, ddfilter.speed

---

**plotMap**

**Plot location data**

**Description**

Function to easily plot locations on a map or a satellite image.

**Usage**

```r
plotMap(sdata, xlim = NULL, ylim = NULL, margin = 10, bgmap = NULL, google.key = NULL, map.bg = "grey", map.col = "black", zoom = "auto", point.bg = "yellow", point.col = "black", point.symbol = 21, point.size = 1, line.col = "lightgrey", line.type = 1, line.size = 0.5, sb.distance = NULL, sb.lwd = 1, sb.line.col = "black", sb.text.size = 4, sb.text.col = "black", sb.space = 3, 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". A map is created for each subset of location data grouped by the unique "id". "DateTime" is date & time in class POSIXct. "lat" and "lon" are the recorded latitude and longitude in decimal degrees.

- **xlim**  
  Limits for x axis. If not specified, the values are determined as the maximum range of the input data with an additional margin (see `margin`).

- **ylim**  
  Limits for x axis. See `xlim` for details.

- **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 provided by the `maps` package is used. The Google Maps ("terrain", "satellite", "roadmap", "hybrid") can also be queried.
google.key

If the 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 of the Google Maps is selected.

map.col

Outline colour of the map. This argument is ignored when any of the Google Maps is selected.

zoom

Map zoom for the Google Maps. 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.

sb.distance

An integer to specify the length of the scale bar. If not specified, approximately a quarter of the plotting range will be used.

sb.lwd

An integer to specify the thickness (width) of the scale bar.

sb.line.col

The colour of the scale bar.

sb.text.size

An integer to specify the text size for the scale bar.

sb.text.col

The colour of the text for the scale bar.

sb.space

Set the amount of space between the scale bar and the text for the scale bar. The value is scaled to the plot. The smaller value increases the space.

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 arrangetlist is returned when multiplot is TRUE. Otherwise a list is returned.

Author(s)

Takahiro Shimada

See Also

dupfilter, ddfilter, est.vmax, est.maxvlp
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 <- est.vmax(turtle.dup)
VLP <- est.maxvlp(turtle.dup)
turtle.dd <- ddfilter(turtle.dup, vmax=V, maxvlp=VLP)
```

#### Plot filtered data for each animal
```
# using the low-resolution world map
plotMap(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
plotMap(turtle.dd, point.size = 2, line.size = 0.5, axes.lab.size = 0, ncol=2, nrow=1,
        bgmap = "satellite", sb.line.col = "white", sb.text.col = "white", key = "an_api_key")
```

```
# End(Not run)
```

---

**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 tide time differences at different places within Sandy Strait.

**Usage**

tidalplane

**Format**

A data frame with 2 rows and 6 variables:

- **secondary.port**  identifier for a station at which tide is only predicted using tidal records observed at the related standard port
- **standard.port**  identifier for a tidal observation station
- **lat**  latitude in decimal degrees
- **lon**  longitude in decimal degrees
- **datumDiff**  baseline difference in meters between bathymetry and tidal observations/predictions if each data uses different datum (e.g. LAT and MSL)
- **timeDiff**  time difference between standard port and its associated secondary port

**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**  date & time in class POSIXct for each observed tidal height
- **reading**  observed tidal height in meters
- **standard.port**  identifier of the tidal station
**turtle**

**Source**

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

---

### 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** date & time in class POSIXct
- **lat** latitude in decimal degrees
- **lon** longitude in decimal degrees
- **qi** numerical quality index associated with each fix where the greater number represents better quality (e.g. number of GPS satellites used for estimation)

**Source**


---

### 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** date & time in class POSIXct
- **lat** latitude in decimal degrees
- **lon** longitude in decimal degrees
- **qi** numerical quality index associated with each fix where the greater number represents better quality (e.g. number of GPS satellites used for estimation)

Source

Index

*Topic datasets
- Australia, 2
- bathymodel, 2
- SandyStrait, 21
- tidalplane, 22
- tidedata, 22
- turtle, 23
- turtle2, 23

Australia, 2

bathymodel, 2

ddfilter, 3, 6, 7, 9, 18–20
ddfilter.loop, 4, 5, 7, 18
ddfilter.speed, 4, 6, 6, 19
depthfilter, 7
distance_filter, 10
dupfilter, 9, 11, 13–16, 20
dupfilter.exact, 12, 12, 14–16
dupfilter.qi, 12, 13, 13, 15, 16
dupfilter.space, 12–14, 14, 16
dupfilter.time, 12–15, 16

est.maxvlp, 4, 6, 17, 20
est.vmax, 4, 7, 18, 20
extract, 8

get_map, 20

linetype, 20

plotMap, 19
POSIXct, 3, 5, 6, 8, 10, 11, 13–19

register_google, 20

SandyStrait, 21
shape, 20

tidalplane, 22