

Package: mort (via r-universe)

November 3, 2024

Title Identifying Potential Mortalities and Expelled Tags in Aquatic Acoustic Telemetry Arrays

Version 0.0.1

Description A toolkit for identifying potential mortalities and expelled tags in aquatic acoustic telemetry arrays. Designed for arrays with non-overlapping receivers.

License GPL (>= 3)

Encoding UTF-8

URL <https://github.com/rosieluain/mort>,
<https://rosieluain.github.io/mort/>

BugReports <https://github.com/rosieluain/mort/issues>

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

LazyData true

Imports lubridate, methods, rlang

Suggests covr, ggplot2, knitr, plotly, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Depends R (>= 3.5.0)

Config/testthat/edition 3

Repository <https://rosieluain.r-universe.dev>

RemoteUrl <https://github.com/rosieluain/mort>

RemoteRef HEAD

RemoteSha 4745211d440eb67d4737c2b23f6f7956ba05a0dc

Contents

backwards	2
ddd	3
detections	4

drift	4
events	6
infrequent	7
morts	10
mortsplot	12
new.data	14
residences	15
resmax	16
resmaxcml	17
review	18
season	20
seasons	22
stationchange	22

Index	25
--------------	-----------

backwards	<i>Shift start time of potential mortalities earlier</i>
-----------	--

Description

Shift the start time of potential mortalities earlier, if station/location has not changed.

Usage

```
backwards(data, morts, ID, station, res.start, stnchange = NULL)
```

Arguments

data	a dataframe of residence events. Residence events must include tag ID, location name, start time, and duration. Residence events should be continuous (i.e., not subset by season).
morts	a dataframe containing potential mortalities. Must use #' the same column names and in the same order as data.
ID	a string of the name of the column in data that holds the tag or sample IDs.
station	a string of the name of the column in data that holds the station name or receiver location.
res.start	a string of the name of the column in data that holds the start date and time. Must be specified and in POSIXt if type="manual".
stnchange	a dataframe with the start time and location of the most recent station or location change. Must use the same column names and in the same order as data.

Value

a dataframe with one row for each tag ID, including the date/time of the residence start when the potential mortality or expelled tag was identified. Returns the input `morts` dataframe if no potential mortalities are shifted earlier. All input data fields (e.g., any name, location, or species information that was included with the input data) will be retained.

Examples

```
morts<-morts(data=events,type="mort",ID="ID",station="Station.Name",
method="any")
head(morts)

# If station change not identified yet:
morts_bw<-backwards(data=events,morts=morts,ID="ID",
station="Station.Name",res.start="ResidenceStart")
head(morts_bw)

# Identify station change first:
station.change<-stationchange(data=events,type="mort",
ID="ID",station="Station.Name")

morts_bw<-backwards(data=events,morts=morts,ID="ID",
station="Station.Name",res.start="ResidenceStart",
stnchange=station.change)
head(morts_bw)
```

ddd

Sample dead drift direction dataframe

Description

The locations have been given generic station names, due to the sensitive nature of the dataset. The connections between stations (i.e., which stations are connected by drift and in which direction) are real.

Usage

```
data(ddd)
```

Format

A data frame with 18 rows and 2 variables.

Details

The variables are as follows:

- From. The station where a tag may drift from.
- To. The station where a tag may drift to, from the station in From.

 detections

Sample acoustic telemetry detection data

Description

These detection data were subset from a real acoustic telemetry dataset of Arctic Char near Kugluktuk, Nunavut. The data were from a project that was a collaboration between the Kugluktuk Hunters and Trappers Organization, the University of Waterloo, and Fisheries and Oceans Canada. The data are valuable to the community of Kugluktuk, and are also sensitive due to their pertinence to the local fishery for Arctic Char. For this reason, the detections have been given generic station names and fish IDs so locations and sample information is anonymous. The year has also been changed, but otherwise, the detections are real, and therefore have all the challenges and intricacies of biological data.

Usage

```
data(detections)
```

Format

A data frame with 447 627 rows (observations) and 3 variables.

Details

The variables are as follows:

- `DateTimeUTC`. The date and time of each detection, in POSIXct format. The data are in UTC, as downloaded from the receivers.
- `Station.Name` Name of the receiver location.
- `ID` Unique ID of the fish.

 drift

Dead and drifting

Description

Identifies sequential residence events where detected movement between stations may be due to drifting of an expelled tag or dead animal.

Usage

```

drift(
  data,
  type,
  ID,
  station,
  res.start = "auto",
  res.end = "auto",
  residences = "auto",
  units = "auto",
  ddd,
  from.station,
  to.station,
  cutoff = NULL,
  cutoff.units = NULL,
  verbose = TRUE
)

```

Arguments

data	a data frame of residence events. Residence events must include tag ID, location name, start time, and end time.
type	the method used to generate the residence events in data. Options are "mort", "actel", "glatos", "vtrack", or "manual". If "manual", then user must specify ID, station, res.start, and res.end.
ID	a string of the name of the column in data that holds the tag or sample IDs.
station	a string of the name of the column in data that holds the station name or receiver location.
res.start	a string of the name of the column in data that holds the start date and time. Must be specified and in POSIXt or character in the format YYYY-mm-dd HH:MM:SS if type="manual".
res.end	a string of the name of the column in data that holds the end date and time. Must be specified and in POSIXt or character in the format YYYY-mm-dd HH:MM:SS if type="manual".
residences	a character string with the name of the column in data that holds the duration of the residence events.
units	units of the duration of the residence events in data.
ddd	a dataframe of stations/locations where detected movement between stations may be due to drifting of an expelled tag or dead animal.
from.station	a string of the name of the column in data that contains the station/location names where drifting detections may start from.
to.station	a string of the name of the column in data that contains the station/location names where drifting detections may move to.
cutoff	the maximum allowable time difference between detections to be considered a single residence event. Default is NULL.

`cutoff.units` the units of the cutoff. Options are "secs", "mins", "hours", "days", and "weeks".
`verbose` option to display progress bar as drift is applied. Default is TRUE.

Value

A data frame with one row for each residence event. Format is the same as the input residence events, but events that may be due to dead drift are combined into single residence events.

Examples

```
# With no drift:
head(events)

drift.events<-drift(data=events[events$ID=="A",],type="mort",ID="ID",
station="Station.Name",ddd=ddd,from.station="From",to.station="To",
verbose=FALSE)
head(drift.events)

# With cutoff:
drift.events<-drift(data=events[events$ID=="A",],type="mort",ID="ID",
station="Station.Name",ddd=ddd,from.station="From",to.station="To",
cutoff=1,cutoff.units="days",verbose=FALSE)
head(drift.events)
```

events	<i>Sample residence events</i>
--------	--------------------------------

Description

Events generated from the sample detection file provided. Events were generated using the `residences()` function in `mort`.

Usage

```
data(events)
```

Format

A data frame with 11 487 rows (events) and 5 variables.

Details

The detection data were subset from a real acoustic telemetry dataset of Arctic Char near Kugluktuk, Nunavut. The data were from a project that was a collaboration between the Kugluktuk Hunters and Trappers Organization, the University of Waterloo, and Fisheries and Oceans Canada. The data are valuable to the community of Kugluktuk, and are also sensitive due to their pertinence to the local fishery for Arctic Char. For this reason, the detections have been given generic station names and fish IDs so locations and sample information is anonymous. The year has also been changed, but

otherwise, the detections are real, and therefore have all the challenges and intricacies of biological data.

The variables are as follows:

- ResidenceStart. The start time of the residence events, in POSIXct.
- Station.Name. Name of the receiver location.
- ID. Unique ID of the fish.
- ResidenceEnd. The end time of the residence events, in POSIXct.
- ResidenceLength.days. The duration of the residence events. The units are days, as indicated by the variable name (which is automatically generated by `mort::residences()`)

infrequent

Identify potential mortalities from infrequent detections

Description

Identifies potential mortalities or expelled tags from infrequent detections in passive acoustic telemetry data. Mortalities are identified based on a user-defined threshold and timeframe.

Usage

```
infrequent(  
  data,  
  type = "mort",  
  ID,  
  station,  
  res.start = "auto",  
  res.end = "auto",  
  residences = "auto",  
  units = "auto",  
  method,  
  threshold,  
  threshold.units = NULL,  
  recent.period = NULL,  
  recent.units = NULL,  
  start = NULL,  
  end = NULL,  
  morts.prev = NULL,  
  replace = FALSE,  
  backwards = FALSE,  
  ddd = NULL,  
  from.station = NULL,  
  to.station = NULL,  
  drift.cutoff = NULL,  
  drift.units = NULL,  
  verbose = TRUE  
)
```

Arguments

<code>data</code>	a dataframe of residence events. Residence events must include tag ID, location name, start time, end time, and duration.
<code>type</code>	the method used to generate the residence events. Options are "mort", "actel", "glatos", "vtrack", or "manual". If "manual", then user must specify ID, station, <code>res.start</code> , <code>res.end</code> , <code>residences</code> , and <code>units</code> .
<code>ID</code>	a string of the name of the column in <code>data</code> that holds the tag or sample IDs.
<code>station</code>	a string of the name of the column in <code>data</code> that holds the station name or receiver location.
<code>res.start</code>	a string of the name of the column in <code>data</code> that holds the start date and time. Must be specified and in POSIXt or character in the format YYYY-mm-dd HH:MM:SS if <code>type="manual"</code> .
<code>res.end</code>	a string of the name of the column in <code>data</code> that holds the end date and time. Must be specified and in POSIXt or character in the format YYYY-mm-dd HH:MM:SS if <code>type="manual"</code> .
<code>residences</code>	<code>residences</code> a character string with the name of the column in <code>data</code> that holds the duration of the residence events.
<code>units</code>	Units of the duration of the residence events in <code>data</code> . Options are "secs", "mins", "hours", "days", and "weeks".
<code>method</code>	a character string of the threshold method. Options are "recent" and "defined". If "recent", must specify <code>recent.period</code> and <code>recent.units</code> . If "defined", must specify <code>start</code> and <code>end</code> .
<code>threshold</code>	the minimum summed duration of residence events for an animal to be considered alive. Units must be the same as the units of <code>residences</code> .
<code>threshold.units</code>	the units of <code>threshold</code> . Options are "secs", "mins", "hours", "days", and "weeks". If NULL, assumed to be the same as <code>units</code> .
<code>recent.period</code>	the length of the period of time in which an animal must be detected longer than the <code>threshold</code> to be considered alive. The period ends with the most recent detection of a given animal.
<code>recent.units</code>	the units of <code>recent.period</code> . Options are "secs", "mins", "hours", "days", and "weeks".
<code>start</code>	character string with the start of the user-defined time period to search for infrequent detections if <code>method="defined"</code> . Must be in the format YYYY-mm-dd HH:MM:SS. The time zone is the same as <code>res.start</code> or assumed to be UTC if no time zone is defined for <code>res.start</code> .
<code>end</code>	character string with the end of the user-defined time period to search for infrequent detections if <code>method="defined"</code> . Must be in the format YYYY-mm-dd HH:MM:SS. The time zone is the same as <code>res.start</code> or assumed to be UTC if no time zone is defined for <code>res.start</code> .
<code>morts.prev</code>	a dataframe containing potential mortalities. The dataframe must have the same columns and in the same order as <code>data</code> .
<code>replace</code>	if <code>morts</code> specified and an animal with infrequent detections is already in <code>morts</code> , the record in <code>morts</code> will be replaced if TRUE. Default is FALSE.

backwards	option to examine residence events prior to the one that was flagged as a potential mortality. If prior residence events are at the same station/location as the flagged event, the time of the potential mortality is shifted earlier.
ddd	an optional dataframe of stations/locations where detected movement between stations may be due to drifting of an expelled tag or dead animal.
from.station	a string of the name of the column in ddd that contains the station/location names where drifting detections may start from. Must be identical to the station/location names in data.
to.station	a string of the name of the column in ddd that contains the station/location names where drifting detections may move to. Must be identical to the station/location names in data.
drift.cutoff	the maximum allowable time difference between detections to be considered a single residence event. Recommended to be the same as used to generate residence events in data.
drift.units	the units of the cutoff. Options are "secs", "mins", "hours", "days", and "weeks". Recommended to be the same as used to generate residence events in data.
verbose	option to display updates and progress bars as sub-functions are called and run. Default is TRUE.

Details

Example of `method="recent"`: if `threshold=10`, `threshold.units="mins"`, `recent.period=52` and `recent.units="weeks"` (1 year), an animal will be flagged as a potential mortality if it was detected for less than 10 minutes within a year, ending with the most recent detection.

Example of `method="defined"`: if `threshold=10`, `threshold.units="mins"`, `start="2019-10-01"`, and `end="2020-06-01"`, an animal will be flagged as a potential mortality if it was detected for less than 10 minutes between 01 October 2019 and 01 June 2020.

Value

if `morts=NULL`, a dataframe with one row for each tag ID, including the date/time of the residence start when the potential mortality or expelled tag was identified. If `morts` is specified, any potential mortalities will be added to existing `morts` dataframe. If `morts` is specified and `replace=TRUE`, then any mortalities that are flagged by `infrequent()` and occurred at an earlier time than those in `morts` will be replaced to the earlier date. All input data fields (e.g., any name, location, or species information that was included with the input data) will be retained.

Examples

```
## Recent example
inf_recent<-infrequent(data=events, type="mort", ID="ID",
  station="Station.Name", method="recent",
  threshold=72, threshold.units="hours",
  recent.period=52, recent.units="weeks",
  verbose=FALSE)
head(inf_recent)

## User-defined example
```

```
inf_defined<-infrequent(data=events,type="mort",ID="ID",
station="Station.Name",method="defined",
threshold=12,threshold.units="hours",
start="2006-06-15",end="2006-10-15",
verbose=FALSE)
head(inf_defined)
```

morts

Identify potential mortalities or expelled tags

Description

Identifies potential mortalities or expelled tags from passive acoustic telemetry data. Mortalities are identified based on thresholds derived from the dataset itself.

Usage

```
morts(
  data,
  type = "mort",
  ID,
  station,
  res.start = "auto",
  res.end = "auto",
  method = "all",
  units = "auto",
  residences = "auto",
  singles = TRUE,
  backwards = FALSE,
  drift = "none",
  ddd = NULL,
  from.station = NULL,
  to.station = NULL,
  drift.cutoff = NULL,
  drift.units = NULL,
  season.start = NULL,
  season.end = NULL,
  season.overlap = TRUE,
  morts.prev = NULL,
  verbose = TRUE
)
```

Arguments

data a dataframe of residence events. Residence events must include tag ID, location name, start time, end time, and duration.

type	the method used to generate the residence events. Options are "mort", "actel", "glatos", "vtrack", or "manual". If "manual", then user must specify ID, station, res.start, res.end, residences, and units.
ID	a string of the name of the column in data that holds the tag or sample IDs.
station	a string of the name of the column in data that holds the station name or receiver location.
res.start	a string of the name of the column in data that holds the start date and time. Must be specified and in POSIXt or character in the format YYYY-mm-dd HH:MM:SS if type="manual".
res.end	a string of the name of the column in data that holds the end date and time. Must be specified and in POSIXt or character in the format YYYY-mm-dd HH:MM:SS if type="manual".
method	the method to be used in flagging mortalities. Options are "last", "any", "cumulative", or "all"
units	units of the duration of the residence events in data.
residences	a character string with the name of the column in data that holds the duration of the residence events.
singles	specifies if single detections (length of residence event = 0) should be retained. Default is TRUE. Note that if single detections are removed (singles=FALSE), backwards will also not include single detections.
backwards	option to examine residence events prior to the one that was flagged as a potential mortality. If prior residence events are at the same station/location as the flagged event, the time of the potential mortality is shifted earlier. Note that if backwards=TRUE, then the output of method="last" is the same as method="any".
drift	option to account for potential drifting in identifying thresholds and/or mortalities. Options are "none", "threshold", "morts", "both". Default is "none".
ddd	a dataframe of stations/locations where detected movement between stations may be due to drifting of an expelled tag or dead animal.
from.station	a string of the name of the column in ddd that contains the station/location names where drifting detections may start from. Must be identical to the station/location names in data.
to.station	a string of the name of the column in ddd that contains the station/location names where drifting detections may move to. Must be identical to the station/location names in data.
drift.cutoff	the maximum allowable time difference between detections to be considered a single residence event. Recommended to be the same as used to generate residence events in data.
drift.units	the units of the cutoff. Options are "secs", "mins", "hours", "days", and "weeks". Recommended to be the same as used to generate residence events in data.
season.start	the start date/time(s) of the period of interest. If the period of interest is the same in all study years, must be a character string in format "dd-mm". Otherwise, must be in POSIXt, or a character string in format YYYY-mm-dd HH:MM:SS.

season.end	the end date/time(s) of the period of interest. If the period of interest is the same in all study years, must be a character string in format "dd-mm". Otherwise, must be in POSIXt, or a character string in format YYYY-mm-dd HH:MM:SS.
season.overlap	option to include residence events that overlap either the beginning or the end of the period of interest. If TRUE, the full overlapping residence events will be retained. If FALSE, only the portion of the residence events that is within the period of interest will be retained, and residences will be recalculated, using specified units. Default is TRUE
morts.prev	a dataframe containing potential mortalities. The dataframe must have the same columns and in the same order as data.
verbose	option to display updates and progress bars as sub-functions are called and run. Default is TRUE.
season	a dataframe with start and end dates of the season(s) of interest

Value

a dataframe with one row for each tag ID, including the date/time of the residence start when the potential mortality or expelled tag was identified. All input data fields (e.g., any name, location, or species information that was included with the input data) will be retained.

Examples

```
morts_ex<-morts(data=events, type="mort", ID="ID",
station="Station.Name", method="any", verbose=FALSE)
head(morts_ex)
```

```
morts_ex_bw<-morts(data=events, type="mort", ID="ID",
station="Station.Name", method="any", backwards=TRUE, verbose=FALSE)
head(morts_ex_bw)
```

mortsplot

Plot residence events

Description

Plot residence events, with the option of plotting identified mortalities. Plotting uses ggplot2. Interactive option also uses plotly.

Usage

```
mortsplot(
  data,
  type,
  ID,
  station,
  res.start = "auto",
  res.end = "auto",
```

```

morts = NULL,
singles = TRUE,
interactive = FALSE,
residences = NULL,
units = NULL,
season.start = NULL,
season.end = NULL,
facet = FALSE,
facet.axis = "x",
facet.by = "season",
verbose = TRUE
)

```

Arguments

data	a dataframe of residence events. Residence events must include tag ID, location name, start time, and end time.
type	the method used to generate the residence events. Options are "mort", "actel", "glatos", "vtrack", or "manual".
ID	a string of the name of the column in data that holds the tag or sample IDs.
station	a string of the name of the column in data that holds the station name or receiver location.
res.start	a string of the name of the column in data that holds the start date and time. Must be specified and in POSIXt or character in the format YYYY-mm-dd HH:MM:SS if type="manual".
res.end	a string of the name of the column in data that holds the end date and time. Must be specified and in POSIXt or character in the format YYYY-mm-dd HH:MM:SS if type="manual".
morts	a dataframe containing potential mortalities. The dataframe must have the same ID, station, res.start, res.end, and residences column names as data.
singles	option to adjust the end times of residence events so single detection events are visible. Adjustment is for visualization purposes only and should not obscure or impact the visualization of other residence events.
interactive	option to generate an interactive plot.
residences	an optional character string with the name of the column in data that holds the duration of the residence events. Required if generating an interactive plot or applying season.
units	units of the duration of the residence events in data. Required if applying season.
season.start	the start date/time(s) of the period of interest. If the period of interest is the same in all study years, must be a character string in format "dd-mm". Otherwise, must be in POSIXt, or a character string in format YYYY-mm-dd HH:MM:SS.
season.end	the end date/time(s) of the period of interest. If the period of interest is the same in all study years, must be a character string in format "dd-mm". Otherwise, must be in POSIXt, or a character string in format YYYY-mm-dd HH:MM:SS.

facet	option to facet by year or season. If TRUE, then season.start and season.end must be provided.
facet.axis	option to position facets along x or y axis. Options are "x" and "y". Default is "x". Note that facet.axis can only be "y" if facet.by="year".
facet.by	option to facet by "season" (as defined with season.start and season.end) or "year". Default is "season".
verbose	option to display updates and progress bar as function is run. Default is TRUE.

Value

a ggplot2 plot. Additional arguments (e.g., formatting axes, legend, aes, manual colour scales) can be added as for any ggplot2 plot. If interactive=TRUE, returns a plotly plot.

Examples

```
plot<-mortsplot(data=events,type="mort",ID="ID",station="Station.Name")
plot

# With mortalities plotted over residences:
morts<-morts(data=events,type="mort",ID="ID",station="Station.Name",
method="any",verbose=FALSE)

plot<-mortsplot(data=events,type="mort",ID="ID",station="Station.Name",
morts=morts)
plot
```

new.data

Example of new acoustic telemetry detection data

Description

These are fabricated detection data to demonstrate the use of the review function.

Usage

```
data(new.data)
```

Format

A data frame with 35 rows (observations) and 3 variables.

Details

The variables are as follows:

- ResidenceStart. The start time of the residence events, in POSIXct.
- Station.Name. Name of the receiver location.

- ID. Unique ID of the fish.
- ResidenceEnd. The end time of the residence events, in POSIXct.
- ResidenceLength.days. The duration of the residence events. The units are days, as indicated by the variable name (which is automatically generated by `mort::residences()`)

residences

Generate residence events

Description

Generate residence events from passive acoustic telemetry data.

Usage

```
residences(data, ID, station, datetime, cutoff, units, verbose = TRUE)
```

Arguments

data	a data frame of detection data.
ID	a string of the name of the column in data that holds the tag or sample IDs.
station	a string of the name of the column in data that holds the station name or receiver location.
datetime	a string of the name of the column in data that holds the date and time.
cutoff	the maximum allowable time difference between detections to be considered a single residence event.
units	the units of the cutoff. These will also be the units used to calculate the duration of the residence events. Options are "secs", "mins", "hours", "days", and "weeks".
verbose	option to display progress bar as residences are generated. Default is TRUE.

Details

Note that a progress bar appears, based on how many of the unique tag IDs have been processed. There will be a delay both before the progress bar appears and after the progress bar has reached 100%, which may be substantial depending on the size of the telemetry dataset.

Value

A data frame with one row for each residence event, including date/time of residence start, date/time of residence end, and duration of residence event. All input data fields (e.g., any name, location, or species information that was included with detection data) will be retained.

Examples

```
head(detections)
res.events<-residences(data=detections[1:500,],ID="ID",station="Station.Name",
datetime="DateTimeUTC",cutoff=1,units="days",verbose=FALSE)
head(res.events)
```

 resmax

Maximum residence duration

Description

Find the maximum duration of a single residence in the dataset that occurred before a station change (i.e., the animal can be assumed to be alive)

Usage

```
resmax(
  data,
  ID,
  station,
  res.start,
  residences,
  stnchange,
  drift = FALSE,
  verbose = TRUE
)
```

Arguments

<code>data</code>	a dataframe of residence events. Residence events must include tag ID, location name, start time, and duration.
<code>ID</code>	a string of the name of the column in <code>data</code> that holds the tag or sample IDs.
<code>station</code>	a string of the name of the column in <code>data</code> that holds the station name or receiver location.
<code>res.start</code>	a string of the name of the column in <code>data</code> that holds the start date and time. Must be specified and in POSIXt if <code>type="manual"</code> .
<code>residences</code>	a character string with the name of the column in <code>data</code> that holds the duration of the residence events.
<code>stnchange</code>	a dataframe with the start time and location of the most recent station or location change. Must use the same column names as <code>data</code> .
<code>drift</code>	indicates if drift residence events should be included in determining the maximum residence duration
<code>verbose</code>	option to display progress bar as function is run. Default is TRUE.

Value

a dataframe with the residence information for the longest residence for each tag ID that occurred before the most recent station/location change.

Examples

```
# Identify most recent station change
station.change<-stationchange(data=events,type="mort",ID="ID",
station="Station.Name",verbose=FALSE)

longest_res_events<-resmax(data=events,ID="ID",station="Station.Name",
res.start="ResidenceStart",residences="ResidenceLength.days",
stnchange=station.change,verbose=FALSE)
head(longest_res_events)
```

resmaxcml

*Maximum cumulative residence duration***Description**

Find the maximum duration that an animal spent at a single station/location before a station change (i.e., the animal can be assumed to be alive). Differs from `resmax` in that the duration is cumulative - the time of residence events and intervals between residence events are all included, provided there are no intervening residence events at other stations/locations.

Usage

```
resmaxcml(
  data,
  ID,
  station,
  res.start,
  res.end,
  residences,
  units,
  stnchange,
  verbose = TRUE
)
```

Arguments

<code>data</code>	a dataframe of residence events. Residence events must include tag ID, location name, start time, and duration.
<code>ID</code>	a string of the name of the column in <code>data</code> that holds the tag or sample IDs.
<code>station</code>	a string of the name of the column in <code>data</code> that holds the station name or receiver location.
<code>res.start</code>	a string of the name of the column in <code>data</code> that holds the start date and time. Must be specified and in POSIXt if <code>type="manual"</code> .
<code>res.end</code>	a string of the name of the column in <code>data</code> that holds the end date and time. Must be specified and in POSIXt if <code>type="manual"</code> .

residences	a character string with the name of the column in data that holds the duration of the residence events.
units	units of the duration of the residence events in data.
stnchange	a dataframe with the start time and location of the most recent station or location change. Must use the same column names as data.
verbose	option to display progress bar as function is run. Default is TRUE.

Value

a dataframe with the cumulative residence information for each period where an animal was consecutively detected at a single station/location. Records are only given for cumulative residences that occurred before the most recent station/location change (i.e., the animal can be assumed to be alive).

Examples

```
# Identify most recent station change
station.change<-stationchange(data=events[events$ID=="A",], type="mort",
ID="ID", station="Station.Name", verbose=FALSE)

cumulative_events<-resmaxcml(data=events[events$ID=="A",], ID="ID",
station="Station.Name", res.start="ResidenceStart", res.end="ResidenceEnd",
residences="ResidenceLength.days", units="days",
stnchange=station.change, verbose=FALSE)
```

review

Review previously identified mortalities using new data

Description

Uses new data to determine if an animal that was previously flagged as a mortality made a station/location change and may therefore be alive.

Usage

```
review(
  morts,
  new.data,
  old.data = NULL,
  type,
  ID,
  station,
  res.start = "auto",
  res.end = NULL,
  residences = NULL,
  units = NULL,
  ddd = NULL,
```

```

    from.station = NULL,
    to.station = NULL,
    verbose = TRUE
  )

```

Arguments

<code>morts</code>	a dataframe with previously flagged mortalities. Format does not need to match <code>new.data</code> exactly, but the names and formats of <code>ID</code> , <code>station</code> , and <code>res.start</code> fields must match in all input dataframes.
<code>new.data</code>	a dataframe of new residence events (i.e., generated from detection data that were not included in earlier mort analyses).
<code>old.data</code>	optional dataframe of residence events that were used in earlier mort analyses. If <code>drift</code> was applied in earlier analyses, including <code>old.data</code> is recommended to avoid falsely identifying station changes.
<code>type</code>	the method used to generate the residence events. Options are "mort", "actel", "glatos", "vtrack", or "manual". If "manual", then user must specify <code>ID</code> , <code>station</code> , <code>res.start</code> , <code>res.end</code> , <code>residences</code> , and <code>units</code> .
<code>ID</code>	a string of the name of the column in <code>morts</code> and <code>new.data</code> that holds the tag or sample IDs.
<code>station</code>	a string of the name of the column in <code>morts</code> and <code>new.data</code> that holds the station name or receiver location.
<code>res.start</code>	a string of the name of the column in <code>morts</code> and <code>new.data</code> that holds the start date and time. Must be specified and in POSIXt or character in the format YYYY-mm-dd HH:MM:SS.
<code>res.end</code>	an optional string of the name of the column in <code>morts</code> and <code>new.data</code> that holds the end date and time. Only needed if <code>drift</code> is applied.
<code>residences</code>	an optional character string with the name of the column in <code>morts</code> and <code>new.data</code> that holds the duration of the residence events. Only needed if <code>drift</code> is applied.
<code>units</code>	optional units of the duration of the residence events in <code>morts</code> and <code>new.data</code> . Only needed if <code>drift</code> is applied.
<code>ddd</code>	a dataframe of stations/locations where detected movement between stations may be due to drifting of an expelled tag or dead animal.
<code>from.station</code>	a string of the name of the column in <code>ddd</code> that contains the station/location names where drifting detections may start from. Must be identical to the station/location names in <code>morts</code> and <code>new.data</code> .
<code>to.station</code>	a string of the name of the column in <code>ddd</code> that contains the station/location names where drifting detections may move to. Must be identical to the station/location names in <code>morts</code> and <code>new.data</code> .
<code>verbose</code>	option to display progress bar as function and called functions are run. Default is TRUE.

Value

A dataframe with one row for each tag ID from `morts` with a station/location change that was identified in `new.data`. The remaining fields will include the information for the residence event that was identified as the station change, so the station change can be verified by the user before removing the animal from `morts`. All input data fields (e.g., any name, location, or species information that was included with the input data) will be retained.

Examples

```
morts<-morts(data=events,type="mort",ID="ID",station="Station.Name",
method="any",verbose=FALSE)
```

```
undead<-review(morts=morts,new.data=new.data,
type="mort",ID="ID",station="Station.Name",verbose=FALSE)
```

season	<i>Select residence events from specified seasons</i>
--------	---

Description

Select residence events from specified seasons, to be used to identify potential mortalities or expelled tags. Useful when animals show strong seasonal patterns in behaviour. For example, a reduction in movement during winter may be falsely identified as a mortality, or increase the threshold use to identify mortalities, which would then cause potential mortalities to be missed.

Usage

```
season(
  data,
  type = "mort",
  ID,
  station,
  res.start = "auto",
  res.end = "auto",
  residences = "auto",
  units = "auto",
  season.start,
  season.end,
  overlap = TRUE,
  verbose = TRUE
)
```

Arguments

data	a dataframe of residence events. Residence events must include tag ID, start time, end time, and duration.
------	--

type	the method used to generate the residence events. Options are "mort", "actel", "glatos", "vtrack", or "manual". If "manual", then user must specify <code>res.start</code> , <code>res.end</code> , <code>residences</code> , and <code>units</code> .
ID	a string of the name of the column in data that holds the tag or sample IDs.
station	a string of the name of the column in data that holds the station name or receiver location.
res.start	a string of the name of the column in data that holds the start date and time. Must be specified and in POSIXt or character in the format YYYY-mm-dd HH:MM:SS if <code>type="manual"</code> .
res.end	a string of the name of the column in data that holds the end date and time. Must be specified and in POSIXt or character in the format YYYY-mm-dd HH:MM:SS if <code>type="manual"</code> .
residences	a character string with the name of the column in data that holds the duration of the residence events.
units	Units of the duration of the residence events in data. Options are "secs", "mins", "hours", "days", and "weeks".
season.start	the start date/time(s) of the period of interest. If the period of interest is the same in all study years, must be a character string in format "dd-mm". Otherwise, must be in POSIXt, or a character string in format YYYY-mm-dd HH:MM:SS.
season.end	the end date/time(s) of the period of interest. If the period of interest is the same in all study years, must be a character string in format "dd-mm". Otherwise, must be in POSIXt, or a character string in format YYYY-mm-dd HH:MM:SS.
overlap	option to include residence events that overlap either the beginning or the end of the period of interest. If TRUE, the full overlapping residence events will be retained. If FALSE, only the portion of the residence events that is within the period of interest will be retained, and <code>residences</code> will be recalculated, using specified <code>units</code> . Default is TRUE.
verbose	option to display updates and progress bars as functions is run. Default is TRUE.

Value

a dataframe in the same format as the input data, with residence events limited to the period(s) of interest.

Examples

```
# Seasons in format dd-mm
season.events<-season(data=events, type="mort", ID="ID",
station="Station.Name", season.start="01-06", season.end="31-10",
verbose=FALSE)
head(season.events)

# Seasons in format YYYY-mm-dd HH:MM:SS
season.start<-c("2003-06-15", "2004-06-21")
season.end<-c("2003-10-15", "2004-10-30")
season.events<-season(data=events, type="mort", ID="ID",
station="Station.Name", season.start=season.start, season.end=season.end, verbose=FALSE)
head(season.events)
```

seasons	<i>Ice-free seasons</i>
---------	-------------------------

Description

The year has been changed, due to the sensitive nature of the dataset. The years have been changed in the same manner as the detection dates, so the ice-free dates correspond to the dataset in the same way as the original dataset.

Usage

```
data(seasons)
```

Format

A data frame with 5 rows (years) and 2 variables.

Details

The variables are as follows:

- Start. The date of ice break-up (the start of the period of interest).
- End. The date of ice freeze-up (the end of the period of interest).

stationchange	<i>Identify most recent station change</i>
---------------	--

Description

Identify the most recent station or location change from passive acoustic telemetry data.

Usage

```
stationchange(  
  data,  
  type = "mort",  
  ID,  
  station,  
  res.start = "auto",  
  res.end = "auto",  
  residences = "auto",  
  singles = TRUE,  
  drift = FALSE,  
  ddd = NULL,  
  units = NULL,
```

```

    from.station = NULL,
    to.station = NULL,
    verbose = TRUE
  )

```

Arguments

data	a dataframe of residence events. Residence events must include tag ID, location name, start time, and duration.
type	the method used to generate the residence events. Options are "mort", "actel", "glatos", "vtrack", or "manual". If "manual", then user must specify ID, station, res.start, and residences.
ID	a string of the name of the column in data that holds the tag or sample IDs.
station	a string of the name of the column in data that holds the station name or receiver location.
res.start	a string of the name of the column in data that holds the start date and time. Must be specified and in POSIXt if type="manual".
res.end	a string of the name of the column in data that holds the end date and time. Must be specified and in POSIXt or character in the format YYYY-mm-dd HH:MM:SS if type="manual".
residences	a character string with the name of the column in data that holds the duration of the residence events.
singles	specifies if single detections (length of residence event = 0) should be retained. Default is TRUE.
drift	option to account for potential drifting in identifying station changes.
ddd	a dataframe of stations/locations where detected movement between stations may be due to drifting of an expelled tag or dead animal.
units	optional units of the duration of the residence events in data. Required if drift=TRUE. Options are "auto", or "secs", "mins", "hours", "days", or "weeks".
from.station	a string of the name of the column in ddd that contains the station/location names where drifting detections may start from. Must be identical to the station/location names in data.
to.station	a string of the name of the column in ddd that contains the station/location names where drifting detections may move to. Must be identical to the station/location names in data.
verbose	option to display progress bar as function is run. Default is TRUE.

Value

a dataframe with one row for each tag ID, including the date/time of the residence start at the most recent station or location, the date/time of the residence end, and duration of the residence event. All input data fields (e.g., any name, location, or species information that was included with the input data) will be retained.

Examples

```
stn.change<-stationchange(data=events,type="mort",ID="ID",station="Station.Name",verbose=FALSE)
head(stn.change)
```


Index

* datasets

- ddd, [3](#)
- detections, [4](#)
- events, [6](#)
- new.data, [14](#)
- seasons, [22](#)

backwards, [2](#)

ddd, [3](#)
detections, [4](#)
drift, [4](#)

events, [6](#)

infrequent, [7](#)

morts, [10](#)
mortsplot, [12](#)

new.data, [14](#)

residences, [15](#)
resmax, [16](#)
resmaxcml, [17](#)
review, [18](#)

season, [20](#)
seasons, [22](#)
stationchange, [22](#)