Package 'HMMoce'

October 29, 2017

Type Package

Title Improved Analysis of Marine Animal Movement Data Using Hidden Markov Models

Version 1.0.0

Date 2017-10-26

Maintainer Camrin Braun <camrin.braun@gmail.com>

Description Improved analysis of marine animal movement data by implementing a state-space hidden Markov model (HMM) to improve position estimates. Position estimates are derived by comparing electronic tag data (from tags deployed on marine animals, typically fish) to three-dimensional oceanographic data.

License MIT + file LICENSE

Depends R (>= 2.10),

Imports dplyr, fields, foreach, imager, locfit, lubridate, maptools, raster, RColorBrewer, rgeos, RNetCDF, sp, parallel, doParallel, curl, methods

Suggests knitr, rmarkdown, png, grid

RoxygenNote 6.0.1

URL http://www.camrinbraun.com/

BugReports https://github.com/camrinbraun/HMMoce/issues

LazyData true

VignetteBuilder knitr

NeedsCompilation no

Author Camrin Braun [aut, cre],
Benjamin Galuardi [aut],
Benjamin Jones [ctb] (Contributed to earlier version of some of the download functions.),
Martin Pedersen [ctb] (Developed an earlier version of some of the HMM framework and helper functions.)

Repository CRAN

Date/Publication 2017-10-29 14:23:39 UTC

R topics documented:

`calc.errEll` *Calculates error ellipses around input locations*

Description

Error ellipses are generated around WC tag based positions using tag-calculated error structure.

Usage

```
calc.errEll(locs, locs.grid)
```

Arguments

`locs`	is data frame of locations typically read from -Locations.csv output from Wildlife Computers
`locs.grid`	is list output from `setup.locs.grid`

Value

an array of error ellipses for input light-based locations

`calc.gpe2` *Calculate Light Likelihood from GPE2 Output*

Description

`calc.gpe2` calculates likelihood estimates for each day of animal tag data.

Usage

```
calc.gpe2(locs, locDates, locs.grid, dateVec, errEll = TRUE, gpeOnly = TRUE)
```

Arguments

`locs`	is data frame from -Locations file output from DAP/Tag Portal for WC tags and contains GPS, Argos, and GPE locations as applicable.
`locDates`	is vector of dates from locs dataframe
`locs.grid`	is list output from `setup.locs.grid`
`dateVec`	is vector of dates from tag to pop-up in 1 day increments.

errEll	is logical indicating whether error ellipses should be generated for light-based likelihoods as given from output of WC-GPE. False if only longitude should be used. If False, standard deviation on light measurements is currently fixed at 0.7 deg longitude following Musyl et al 2011. Default is FALSE and will use longitude only.
gpeOnly	is logical. If TRUE (default), locs input is trimmed to GPE positions only. This is most applicable in scenarios with FastGPS data and you're adding this as a GPS input.

Details

Light errors are parameterized using elliptical error values output in '-Locations.csv' (WC tags).

Value

L is an array of lon x lat likelihood surfaces (matrices) for each time point (3rd dimension)

References

Musyl MK, Domeier ML, Nasby-Lucas N, Brill RW, McNaughton LM, Swimmer JY, Lutcavage MS, Wilson SG, Galuardi B, Liddle JB (2011) Performance of pop-up satellite archival tags. Mar Ecol Prog Ser

See Also

calc.srss

Examples

```
## Setup for calculating light likelihood
# Read the data
locsFile <- system.file("extdata", "141259-Locations-GPE2.csv", package = "HMMoce")
locs <- read.table(locsFile, sep = ',', header = TRUE, blank.lines.skip = FALSE)

# Set spatial and temporal limits
sp.lim <- list(lonmin = -82, lonmax = -25, latmin = 15, latmax = 50)
locs.grid <- setup.locs.grid(sp.lim)
iniloc <- data.frame(matrix(c(13, 10, 2015, 41.3, -69.27, 10, 4, 2016, 40.251, -36.061),
 nrow = 2, ncol = 5, byrow = TRUE))
names(iniloc) <- list('day','month','year','lat','lon')
tag <- as.POSIXct(paste(iniloc[1,1], '/', iniloc[1,2], '/', iniloc[1,3], sep=''),
format = '%d/%m/%Y', tz='UTC')
pop <- as.POSIXct(paste(iniloc[2,1], '/', iniloc[2,2], '/', iniloc[2,3], sep=''),
format = '%d/%m/%Y', tz='UTC')
dateVec <- as.Date(seq(tag, pop, by = 'day'))

# Try a calculation
L.light <- calc.gpe2(locs[1,], iniloc, locs.grid, dateVec, errEll=TRUE, gpeOnly=TRUE)

## Not run:
# Full example light calculation
```

```
L.light <- calc.gpe2(locs, iniloc = iniloc, locs.grid = locs.grid,
                     dateVec = dateVec, errEll = TRUE, gpeOnly = TRUE)

## End(Not run)
```

`calc.hycom` *Hycom Profile LIkelihood*

Description

Calculate Hycom profile likelihood surface

Usage

```
calc.hycom(pdt, filename, hycom.dir, focalDim = 9, dateVec, use.se = TRUE)
```

Arguments

`pdt`	input PDT data output from `read.wc` and `extract.pdt`
`filename`	is the first part of the filename specified to the download function `get.env`. For example, if downloaded files were specific to a particular dataset, you may want to identify that with a name like 'tuna' or 'shark1'. This results in a downloaded filename of, for example, 'tuna_date.nc'. This filename is required here so the calc function knows where to get the env data.
`hycom.dir`	directory of downloaded hycom (or other) data
`focalDim`	is integer for dimensions of raster::focal used to calculate sd() of temperature grid cell. Recommend focalDim = 9 for Hycom data at 0.08deg resolution.
`dateVec`	vector of complete dates (from tag to pop by day) for data range. This should be in 'Date' format
`use.se`	is logical indicating whether or not to use SE when using regression to predict temperature at specific depth levels.

Value

a raster brick of Hycom profile likelihood

See Also

`calc.hycom.par`

calc.hycom.par *Hycom Profile LIkelihood in Parallel*

Description

Calculate Hycom profile likelihood surface in parallel

Usage

```
calc.hycom.par(pdt, filename, hycom.dir, focalDim = 9, dateVec,
  use.se = TRUE, ncores = NULL)
```

Arguments

pdt	input PDT data output from read.wc and extract.pdt
filename	is the first part of the filename specified to the download function get.env. For example, if downloaded files were specific to a particular dataset, you may want to identify that with a name like 'tuna' or 'shark1'. This results in a downloaded filename of, for example, 'tuna_date.nc'. This filename is required here so the calc function knows where to get the env data.
hycom.dir	directory of downloaded hycom (or other) data
focalDim	is integer for dimensions of raster::focal used to calculate sd() of temperature grid cell. Recommend focalDim = 9 for Hycom data at 0.08deg resolution.
dateVec	vector of complete dates (from tag to pop by day) for data range. This should be in 'Date' format
use.se	is logical indicating whether or not to use SE when using regression to predict temperature at specific depth levels.
ncores	specify number of cores, or leave blank and use whatever you have!

Value

a raster brick of Hycom profile likelihood

calc.locs *Calculate Position-based Likelihood*

Description

calc.locs calculates likelihood estimates for each day of animal tag data using tag-based locations.

Usage

```
calc.locs(locs, gps = NULL, iniloc, locs.grid, dateVec)
```

Arguments

locs	is data frame from -Locations file output from DAP/Tag Portal for WC tags and contains GPS, Argos, and GPE locations as applicable.
gps	is data frame from -FastGPS file output from WC Tag Portal
iniloc	is 2 x 5 dataframe containing day, month, year, lat, lon for both tag and pop locations
locs.grid	is list output from setup.locs.grid
dateVec	is vector of dates from tag to pop-up in 1 day increments.

Details

GPS and Argos positions are given a "likelihood" using this function but are currently both considered to be "known" positions without error.

Value

L is an array of lon x lat likelihood surfaces (matrices) for each time point (3rd dimension)

---

calc.ohc *Calculate Ocean Heat Content (OHC) likelihood surface*

Description

Compare tag data to OHC grid and calculate likelihoods

Usage

```
calc.ohc(pdt, filename, isotherm = "", ohc.dir, dateVec, bathy = TRUE,
  use.se = TRUE)
```

Arguments

pdt	input PDT data see extract.pdt
filename	is the first part of the filename specified to the download function get.env. For example, if downloaded files were specific to a particular dataset, you may want to identify that with a name like 'tuna' or 'shark1'. This results in a downloaded filename of, for example, 'tuna_date.nc'. This filename is required here so the calc function knows where to get the env data.
isotherm	default " in which isotherm is calculated on the fly based on daily tag data. Otherwise, numeric isotherm constraint can be specified (e.g. 20 deg C).
ohc.dir	directory of downloaded hycom (or other) data
dateVec	is vector of dates from tag to pop-up in 1 day increments.
bathy	is logical indicating whether or not a bathymetric mask should be applied
use.se	is logical indicating whether or not to use SE when using regression to predict temperature at specific depth levels.

Value

likelihood is raster brick of likelihood surfaces representing estimated position based on tag-based OHC compared to calculated OHC using HYCOM

References

Luo J, Ault JS, Shay LK, Hoolihan JP, Prince ED, Brown C a., Rooker JR (2015) Ocean Heat Content Reveals Secrets of Fish Migrations. PLoS One 10:e0141101

See Also

calc.woa

---

calc.ohc.par *OHC Likelihood in Parallel*

Description

Calculate Ocean Heat Content (OHC) likelihood surface in parallel

Usage

```
calc.ohc.par(pdt, filename, isotherm = "", ohc.dir, dateVec, bathy = TRUE,
  use.se = TRUE, ncores = NULL)
```

Arguments

pdt	input PDT data see extract.pdt
filename	is the first part of the filename specified to the download function get.env. For example, if downloaded files were specific to a particular dataset, you may want to identify that with a name like 'tuna' or 'shark1'. This results in a downloaded filename of, for example, 'tuna_date.nc'. This filename is required here so the calc function knows where to get the env data.
isotherm	default " in which isotherm is calculated on the fly based on daily tag data. Otherwise, numeric isotherm constraint can be specified (e.g. 20 deg C).
ohc.dir	directory of downloaded hycom (or other) data
dateVec	vector of complete dates for data range. This should be in 'Date' format
bathy	is logical indicating whether or not a bathymetric mask should be applied
use.se	is logical indicating whether or not to use SE when using regression to predict temperature at specific depth levels.
ncores	specify number of cores, or leave blank and use whatever you have!

Value

a raster brick of OHC likelihood

References

Luo J, Ault JS, Shay LK, Hoolihan JP, Prince ED, Brown C a., Rooker JR (2015) Ocean Heat Content Reveals Secrets of Fish Migrations. PLoS One 10:e0141101

See Also

calc.ohc

`calc.param` *Calculates movement parameters used for behavior kernels*

Description

`calc.param` calculates movement parameters used to generate kernels

Usage

```
calc.param(migr.spd, resid.frac = 0.1, g)
```

Arguments

`migr.spd`	is numeric indicating movement speed of animal (in m s) while in migratory behavior state.
`resid.frac`	is numeric indicating percent of migratory speed that should be used for resident-like movements. Default is 10% of migratory speed.
`g`	is grid from `setup.grid`

Details

Movement parameters for each of two behavioral modes are generated using an input grid and user-input speed (m s) for at least the migratory state.

Value

a list of movement parameters

`calc.srss` *Calculate Position-based Likelihood from SRSS*

Description

`calc.srss` calculates likelihood estimates for each day's estimated dawn dusk times from the tag

Usage

```
calc.srss(light = NULL, locs.grid, dateVec, res = 1, focalDim = 3)
```

Arguments

`light`	is data frame from -LightLoc file output from DAP/Tag Portal for WC tags and contains tag-measured dawn/dusk times.
`locs.grid`	is list output from `setup.locs.grid`
`dateVec`	is vector of dates from tag to pop-up in 1 day increments.
`res`	is resolution of light grid in degrees. default is 1 deg. higher resolution (e.g. res = .25 for 1/4 deg) takes considerably longer to compute
`focalDim`	is integer for dimensions of raster::focal used to calculate sd() of SRSS grid cell. Recommend focalDim = 3 if res=1 and focalDim = 9 if res=0.25.

Details

Tag-measured sunrise/sunset (SRSS) times are first filtered according to the min/max times possible. possible values are computed based on spatial limits included in locs.grid input. after filtering, each yearday has a grid of georeferenced SRSS times that the tag-measured times are compared to in order to generate a likelihood.

Value

L is a raster of dim(lon x lat x dateVec) containing likelihood surfaces for each time point

See Also

`calc.gpe2`

calc.sst *Calculate SST-based likelihood*

Description

calc.sst compares tag SST to remotely sensed SST and calculates likelihoods

Usage

```
calc.sst(tag.sst, filename, sst.dir, dateVec, focalDim = NULL, sens.err = 1)
```

Arguments

tag.sst	is data frame containing tag-collected SST data
filename	is the first part of the filename specified to the download function get.env. For example, if downloaded files were specific to a particular dataset, you may want to identify that with a name like 'tuna' or 'shark1'. This results in a downloaded filename of, for example, 'tuna_date.nc'. This filename is required here so the calc function knows where to get the env data.
sst.dir	local directory where remote sensing SST downloads are stored
dateVec	is vector of dates from tag to pop-up in 1 day increments.
focalDim	is integer for dimensions of raster::focal used to calculate sd() of env grid cell. If left to NULL (default), this dimension will approximately incorporate 0.25 degrees.
sens.err	is numeric indicating the percent sensor error in the tag sst sensor. Default is 1.

Value

likelihood is raster brick of likelihood surfaces representing matches between tag-based sst and remotely sensed sst maps

See Also

calc.ohc

calc.sst.par *Calculate SST-based likelihood in parallel*

Description

calc.sst.par compares tag SST to remotely sensed SST and calculates likelihoods in parallel

Usage

```
calc.sst.par(tag.sst, filename, sst.dir, dateVec, focalDim = NULL,
  sens.err = 1, ncores = NULL)
```

Arguments

tag.sst	is data frame containing tag-collected SST data
filename	is the first part of the filename specified to the download function get.env. For example, if downloaded files were specific to a particular dataset, you may want to identify that with a name like 'tuna' or 'shark1'. This results in a downloaded filename of, for example, 'tuna_date.nc'. This filename is required here so the calc function knows where to get the env data.
sst.dir	local directory where remote sensing SST downloads are stored
dateVec	is vector of dates from tag to pop-up in 1 day increments.
focalDim	is integer for dimensions of raster::focal used to calculate sd() of env grid cell. If left to NULL (default), this dimension will approximately incorporate 0.25 degrees.
sens.err	is numeric indicating the percent sensor error in the tag sst sensor. Default is 1.
ncores	is integer indicating number of cores used in this parallel computation. Defaults to using a detection function that chooses cores for you.

Value

likelihood is raster brick of likelihood surfaces representing matches between tag-based sst and remotely sensed sst maps

See Also

calc.sst

---

calc.track *Calculate most probable track from state estimates*

Description

calc.track uses HMM output via hmm.smoother to calculate most probable track and behavior state

Usage

```
calc.track(distr, g, dateVec, iniloc, method = "mean")
```

Arguments

distr	is output array from hmm.smoother
g	is one of the outputs from resample.grid which denotes what spatial scale and grid you're working on
dateVec	is vector of dates from tag to pop-up in 1 day increments.
iniloc	is matrix of tag and pop locations. Default is NULL because this should be taken care of elsewhere.
method	is character indicating what method to use for track calculation. Currently only 'mean' and 'max' are supported.

Value

calculated track

Examples

```
## Not run:

# GET THE MOST PROBABLE TRACK
tr <- calc.track(s, g, dateVec, iniloc)


## End(Not run)
```

---

calc.woa *Calculate WOA profile in parallel*

Description

Calculate Depth-temperature profile based likelihood

Usage

```
calc.woa(pdt, ptt, woa.data = NULL, dateVec, sp.lim = NULL,
  focalDim = NULL, use.se = TRUE)
```

Arguments

pdt	input PDT data output from read.wc and extract.pdt
ptt	is unique tag identifier
woa.data	is (typically) a list of monthly global 1/4deg climatology data from WOA13. See get.env.
dateVec	is vector of dates from tag to pop-up in 1 day increments.
sp.lim	is list of limits as list(xmin, xmax, ymin, ymax)

focalDim	is integer for dimensions of raster::focal used to calculate sd() of temperature grid cell. Recommend focalDim = 3 if woa.data = woa.one and 9 if using woa.quarter.
use.se	is logical indicating whether or not to use SE when using regression to predict temperature at specific depth levels.

Details

calc.woa.par calculates likelihood of animal position based on summarized depth-temperature profiles

Tag-based depth-temperature profile summaries are compared to climatological profiles from the World Ocean Atlas (WOA) "matched" to generate position likelihoods. This essentially attempts to estimate animal position based on the water mass it is in, particularly if extensive diving performs thorough sampling of the environment. However, remember the in situ data is being compared to climatological means or the results of an oceanographic model.

Value

raster brick of likelihood

See Also

calc.ohc

---

calc.woa.par *Calculate WOA profile likelihood in parallel*

Description

Calculate Depth-temperature profile based likelihood

Usage

```
calc.woa.par(pdt, ptt, woa.data = NULL, dateVec, sp.lim = NULL,
  focalDim = NULL, use.se = TRUE, ncores = NULL)
```

Arguments

pdt	input PDT data output from read.wc and extract.pdt
ptt	is unique tag identifier
woa.data	is (typically) a list of monthly global 1/4deg climatology data from WOA13. See get.env.
dateVec	is vector of dates from tag to pop-up in 1 day increments.
sp.lim	is list of limits as list(xmin, xmax, ymin, ymax)

focalDim	is integer for dimensions of raster::focal used to calculate sd() of temperature grid cell. Recommend focalDim = 3 if woa.data = woa.one and 9 if using woa.quarter.
use.se	is logical indicating whether or not to use SE when using regression to predict temperature at specific depth levels.
ncores	is integer indicating number of cores used in this parallel computation. Defaults to using a detection function that chooses cores for you.

Details

calc.woa.par calculates likelihood of animal position based on summarized depth-temperature profiles

Tag-based depth-temperature profile summaries are compared to climatological profiles from the World Ocean Atlas (WOA) "matched" to generate position likelihoods. This essentially attempts to estimate animal position based on the water mass it is in, particularly if extensive diving performs thorough sampling of the environment. However, remember the in situ data is being compared to climatological means (WOA) or the results of an oceanographic model (HYCOM).

Value

raster brick of likelihood

See Also

calc.ohc

---

expmax *Expectation-maximization framework for state-switching*

Description

expmax performs expectation-maximization for state switching probability

Usage

```
expmax(p.init, g, L, K1, K2, niter = 1000, threshold = 0.01, save = F)
```

Arguments

p.init	a vector of length 2. The first is the probability of staying in behavior state 1 if currently in state 1 and the second for staying in state 2.
g	grid from setup.grid
L	final likelihood
K1	first movement (diffusion) kernel see gausskern
K2	second movement (diffusion) kernel see gausskern

niter	is integer that determines number of iterations to perform
threshold	is threshold of percent change that we consider satisfactory for convergence. Default is 1%.
save	is logical indicating whether the function should save and return a 2 col dataframe of the iterations it went through before crossing the convergence threshold. Defaults to not saving.

Details

Light errors are parameterized using elliptical error values output in.

Value

a 2x2 matrix of state switching probabilities. See P.init input for more information.

References

Woillez M, Fablet R, Ngo TT, et al. (2016) A HMM-based model to geolocate pelagic fish from high-resolution individual temperature and depth histories: European sea bass as a case study. Ecol Modell 321:10-22.

Examples

```
# GENERATE MOVEMENT KERNELS. D VALUES ARE MEAN AND SD PIXELS
K1 <- gausskern(3, 1, muadv = 0)
K2 <- gausskern(10, 5, muadv = 0)

# MAKE A GUESS AT STATE SWITCHING PROBABILITY
# probability of staying in state 1 and 2, respectively
p.init <- c(0.7, 0.8)

## Not run:
# Not run as it relies on L, a large likelihood grid
# RUN EXPECTATION-MAXIMIZATION ROUTINE FOR MATRIX, P (STATE SWITCH PROBABILITY)
P.final <- expmax(p.init, g = g, L = L, K1, K2)


## End(Not run)
```

---

extract.pdt *Extract PDT from Wildlife Computers tag data*

Description

extract.pdt is a simple formatting function that parses PDT data and makes it usable to subsequent functions

Usage

```
extract.pdt(pdt)
```

Arguments

pdt	data frame read from -PDTs.csv output of Wildlife Computers DAP processor or Tag Portal.

Value

data frame formatted for pdt data

---

extract.woa *Extract temperatures from World Ocean Atlas*

Description

extract.woa extracts the desired temperature data from a global dataset derived from monthly gridded climatology data contained in the 2013 World Ocean Atlas

Usage

```
extract.woa(dir, bbox = NULL, resolution)
```

Arguments

dir	path to load the global nc file from; specify the complete path to the nc file unless it is in your current working directory
bbox	bounding box of form list(long min, long max, lat min, lat max)
resolution	indicates whether oceanographic data is gridded at 'quarter' or 'one' degree resolution

Value

a list containing: DAT is an array of temperature data with dimensions (long, lat, depth, time) depth contains 57 standard depth levels by default and levels are defined in variable 'depth' contained here. time is monthly and spans the entire year. LON/LAT are vectors of lon/lat bounds

Examples

```
## Not run:
# download WOA data (large RData file!) from GitHub
woa.dir <- paste('my_woa_dir')
get.env(type = 'woa', resol = 'quarter')

# extract desired data based on spatial bounds
woa <- extract.woa(woa.dir, bbox, 'quarter')
```

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

`findDateFormat` *Determine date format of vector*

Description

`findDateFormat` determines the date format of a given vector of dates

Usage

```
findDateFormat(dateVec)
```

Arguments

`dateVec`	a character vector representing dates

Value

dateformat is character string used as input to strptime(format = dateformat)

Examples

```
dte <- '2015-01-01 05:30:17'
findDateFormat(dte)
dte.POSIX <- as.POSIXct(dte, format = findDateFormat(dte))
dte.POSIX
```

`gausskern` *Create Gaussian Kernel*

Description

`gausskern` calculates 2D Gaussian kernel based on kernel size, deviation, and advection

Usage

```
gausskern(siz, sigma, muadv = 0)
```

Arguments

`siz`	size of the kernel, siz x siz. Must be a positive integer.
`sigma`	standard deviation of the kernel. Unit is cell width. Must be a positive number.
`muadv`	advection of the kernel. Unit of the input is cell width. Defaults to 0.

Value

Gaussian kernel as a 2D matrix of size (siz x siz)

Author(s)

Function originally written for Matlab by Martin W. Pedersen, translated to R by Benjamin Galuardi

References

Pedersen, M.W., Righton, D., Thygesen, U.H., Andersen, K.H., and Madsen, H. 2008. Geolocation of North Sea cod (Gadus morhua) using hidden Markov models and behavioural switching. Canadian Journal of Fisheries and Aquatic Sciences 65(11): 2167-1377.

Examples

```
kern = gausskern(3, 0.5)
```

---

get.bath.data *Download bathymetry data*

Description

Download ETOPO bathymetry. Resolution is either 30 second or one minute resolution

Usage

```
get.bath.data(lonlow, lonhigh, latlow, lathigh, folder = tempdir(),
  seaonly = T, res = c(0.5), raster = TRUE)
```

Arguments

lonlow	numeric indicating minimum longitude extent of desired download (-180 to 180).
lonhigh	see lonlow
latlow	see lonlow
lathigh	see lonlow
folder	is destination folder. Default is a temporary directory.
seaonly	is logical indicating whether you want to download only bathymetry below sea level.
res	is numeric indicating resolution in minutes. Choices currently are 0.5 or 1minute.
raster	is logical indicating whether you want the function to return a raster or not (a list will be returned).

Value

Downloads a NetCDF file containing ETopo bathymetry. If raster=TRUE, a raster is generated from the downloaded NetCDF. Otherwise, the file is just downloaded.

Note

Be patient! The download can take a few minutes!

Examples

```
## Not run:
# Not run to prevent actual data download
sp.lim <- list(lonmin = -82, lonmax = -25, latmin = 15, latmax = 50)
bathy <- get.bath.data(sp.lim$lonmin, sp.lim$lonmax, sp.lim$latmin,
sp.lim$latmax, folder = tempdir())

## End(Not run)
```

---

get.env *Download and Read Oceanographic Data*

Description

`get.env` accesses oceanographic data like sea surface temperature from a remote server and downloads the temporal and spatial extent of interest for further use

Usage

```
get.env(uniqueDates = NULL, filename = NULL, type = NULL,
  spatLim = NULL, resol = NULL, save.dir = getwd(), sst.type = NULL,
  depLevels = NULL)
```

Arguments

`uniqueDates`	is a POSIXct vector of desired dates
`filename`	is first part of the filename specified to the download function. For example, if downloaded files were specific to a particular dataset, you may want to identify that with a name like 'tuna' or 'shark1'. This results in a downloaded filename of, for example, 'tuna_date.nc'.
`type`	is a character string indicating whether you're after sea surface temperature 'sst', hybrid coordinate ocean model 'hycom', or world ocean atlas 'woa' data
`spatLim`	is a list of spatial limits as `list(xmin, xmax, ymin,ymax)`
`resol`	is character describing the desired resolution in degrees if type = 'woa', otherwise NULL. Choices are 'one' or 'quarter'.
`save.dir`	is the directory to save the downloaded data to

sst.type	is character indicating type of desired SST product. Choices are currently Optimum Interpolation ('oi') https://www.ncdc.noaa.gov/oisst or a high-resolution composite ('ghr') https://www.ghrsst.org/.
depLevels	is an integer describing which depth levels to download from Hycom (e.g. 1=surface). Default is NULL and all levels are downloaded.

Value

nothing, just downloads the data to your local machine

Examples

```
## Not run:
# Not run to prevent actual data download
sp.lim <- list(lonmin = -82, lonmax = -25, latmin = 15, latmax = 50)
# FOR OI SST DATA
get.env(as.Date('2015-10-01'), filename='oisst', type = 'sst',
sst.type='oi', spatLim = sp.lim, save.dir = tempdir())

# FOR HYCOM DATA
get.env(as.Date('2015-10-01'), filename='hycom', type = 'hycom',
spatLim = sp.lim, save.dir = tempdir())

# FOR WORLD OCEAN ATLAS DATA
get.env(type = 'woa', resol = 'quarter', save.dir = woa.dir)

## End(Not run)
```

---

get.ghr.sst *Download GHR Sea Surface Temperature (SST) data*

Description

get.ghr.sst downloads sea surface temperature (SST) data for given temporal and spatial constraints of your data.

Usage

```
get.ghr.sst(limits, time, filename = "", download.file = TRUE,
  dir = getwd())
```

Arguments

limits	A list of length 4; minlon, maxlon, minlat, maxlat. Longitude values are -180,180
time	A vector of length 2 with the minimum and maximum times in form as.Date(date).
filename	An optional filename. If provided, then the data is downloaded to that file. Otherwise the data is not downloaded and the url is returned.

download.file	Logical. Should use the default download.file to query the server and download or use the optional curl. Some users may need to use curl in order to get this to work.
dir	is local directory where ncdf files should be downloaded to. default is current working directory. if enter a directory that doesnt exist, it will be created.

Details

The method may return before the download is completed. It will continue to display progress until the download completes. Change the default download.file if the download is failing on your platform.

Value

The url used to extract the requested data from the NetCDF subset service.

Author(s)

Function originally written for R by Ben Jones (WHOI) and modified by Camrin Braun and Ben Galuardi.

References

https://www.ncdc.noaa.gov/oisst

---

get.hycom *Download HYCOM data*

Description

get.hycom downloads HYbrid Coordinate Ocean Model (HYCOM) data for given temporal and spatial constraints of your data.

Usage

```
get.hycom(limits, time, vars = c("water_temp"), include_latlon = TRUE,
  filename = "", download.file = TRUE, dir = getwd(), depLevels = NULL)
```

Arguments

limits	A list of length 4; minlon, maxlon, minlat, maxlat. Longitude values are -180,180
time	A vector of length 2 with the minimum and maximum times in form as.Date(date).
vars	A list of variables to download. This can contain 'water_temp', 'water_u', 'water_v', 'salinity' or 'surf_el' but is not checked for errors.
include_latlon	Should the array of latitude and longitude values be included?

filename	An optional filename. If provided, then the data is downloaded to that file. Otherwise the data is not downloaded and the url is returned.
download.file	Logical. Should use the default download.file to query the server and download or use the optional curl. Some users may need to use curl in order to get this to work.
dir	is local directory where ncdf files should be downloaded to. default is current working directory. if enter a directory that doesn't exist, it will be created.
depLevels	is an integer describing which depth levels to download from Hycom (e.g. 1=surface). Default is NULL and all levels are downloaded.

Details

The method may return before the download is completed. It will continue to display progress until the download completes. Change the default download.file if the download is failing on your platform.

Value

The url used to extract the requested data from the NetCDF subset service.

Author(s)

Function originally written for R by Ben Jones (WHOI) and modified by Camrin Braun and Ben Galuardi.

References

https://hycom.org/

---

get.nll.fun *Negative Log likelihood of parameters*

Description

Negative Log likelihood of parameters

Usage

```
get.nll.fun(parvec = c(10, 30, 5, 2, 0.707, 0.8), g, L)
```

Arguments

parvec	vector of length 6 containing * kernel 1 * kernel 2 * diagonal of 2x2 matrix
g	grid from setup.grid
L	final likelihood (2D)

Value

parameter values

References

Pedersen MW, Patterson TA, Thygesen UH, Madsen H (2011) Estimating animal behavior and residency from movement data. Oikos 120:1281-1290. doi: 10.1111/j.1600-0706.2011.19044.x

---

`get.oi.sst` *Download OI Sea Surface Temperature (SST) data*

Description

`get.oi.sst` downloads sea surface temperature (SST) data for given temporal and spatial constraints of your data.

Usage

```
get.oi.sst(limits, time, filename = "", download.file = TRUE,
  dir = getwd())
```

Arguments

`limits`	A list of length 4; minlon, maxlon, minlat, maxlat. Longitude values are -180,180
`time`	A vector of length 2 with the minimum and maximum times in form `as.Date(date)`.
`filename`	An optional filename. If provided, then the data is downloaded to that file. Otherwise the data is not downloaded and the url is returned.
`download.file`	Logical. Should use the default `download.file` to query the server and download or use the optional `curl`. Some users may need to use `curl` in order to get this to work.
`dir`	is local directory where ncdf files should be downloaded to. default is current working directory. if enter a directory that doesn't exist, it will be created.

Details

The method may return before the download is completed. It will continue to display progress until the download completes. Change the default `download.file` if the download is failing on your platform.

Value

The url used to extract the requested data from the NetCDF subset service.

Author(s)

Function originally written for R by Ben Jones (WHOI) and modified by Camrin Braun and Ben Galuardi.

References

https://www.ncdc.noaa.gov/oisst

---

get.woa *Download World Ocean Atlas Climatology*

Description

get.woa downloads World Ocean Atlas 2013 mean climatological temperature data from GitHub

Usage

```
get.woa(save.dir = getwd(), resol = "one")
```

Arguments

save.dir	is the directory to save the downloaded data to
resol	is character describing the desired resolution in degrees. Choices are 'one' or 'quarter'.

Value

name and directory of the downloaded data

---

getCtr *Get specified contour coordinates*

Description

getCtr uses hmm.smoother output to calculate coordinates at a specified contour of each posterior distribution of the state

Usage

```
getCtr(distr, tr, g, threshold = 50, makePlot = FALSE)
```

Arguments

distr	is output array from hmm.smoother
tr	is output dataframe from calc.track
g	grid from setup.grid
threshold	numeric indicating the percent contour of interest. Default is 50 percent.
makePlot	is logical indicating whether or not to plot at each iteration

Value

a list of length T, dim(distr)[2], containing 1) coordinates of the contour at each time, t. 2) the x and y distance to that contour from the mean of the distribution (lat/lon in track). 3) reference coordinate from mean of distribution. 4) the points of intersection of the contour by which distance is measured in x and y

---

hmm.filter *HMM filter functions*

Description

HMM filter functions

Usage

```
hmm.filter(g, L, K1, K2, P, maskL = T, bound.thr = 0.1, minBounds = 10)
```

Arguments

g	grid from setup.grid
L	is likelihood array output from make.L
K1	first movement (diffusion) kernel see gausskern
K2	second movement (diffusion) kernel see gausskern
P	2x2 probability matrix for transitions between states (K1 and K2)
maskL	is logical indicating whether to mask the input L layer. See mask.L for details.
bound.thr	is numeric indicating the percent threshold that is added and subtracted from the bounding box of the filter output from the previous day before masking. Default is .05 (5 percent).
minBounds	is size (in degrees) of the minimum bounding box around the previous days filter prediction that L data within that box will be included. Outside this box (centered on t-1 filter prediction), L will be masked out.

Value

a list: list(phi = phi, pred = pred, psi = psi) where

- phi. is the probability for each state at each time step
- pred. is ....
- psi. is....

References

Pedersen MW, Patterson TA, Thygesen UH, Madsen H (2011) Estimating animal behavior and residency from movement data. Oikos 120:1281-1290. doi: 10.1111/j.1600-0706.2011.19044.x

Examples

```
## Not run:
# Not run as function relies on large arrays of likelihoods
# RUN THE FILTER STEP
f <- hmm.filter(g, L, K1, K2, maskL=T, P.final, minBounds = bnd)
nllf <- -sum(log(f$psi[f$psi>0])) # negative log-likelihood


## End(Not run)
```

`hmm.smoother` *Smoother recursion over filtered state estimates*

Description

`hmm.smoother` provides backward (starting at end) recursion over filtered state estimates as output from `hmm.filter`. The product of this function an array containing final state estimates.

Usage

```
hmm.smoother(f, K1, K2, L, P)
```

Arguments

`f`	is array output from `hmm.filter`
`K1`	is movement kernel generated by `gausskern` for behavior state 1
`K2`	is movement kernel generated by `gausskern` for behavior state 2
`L`	is likelihood array output from `make.L`
`P`	is transition matrix (usually 2x2) representing probability of state switching

Value

an array of the final state estimates of dim(state, time, lon, lat)

References

Pedersen MW, Patterson TA, Thygesen UH, Madsen H (2011) Estimating animal behavior and residency from movement data. Oikos 120:1281-1290. doi: 10.1111/j.1600-0706.2011.19044.x

Examples

```
## Not run:
# Not run as function relies on large arrays of likelihoods
# RUN THE SMOOTHING STEP
s <- hmm.smoother(f, K1, K2, L, P.final)

## End(Not run)
```

`HMMoce` *HMMoce: an R package for improved analysis of marine animal movement data using hidden Markov models*

Description

The HMMoce package provides a workflow for leveraging all available data from tags deployed on marine animals to estimate movements, space use, and behavior. Marine animals, mostly fish, are notoriously difficult to track with electronic tags because these devices require occupation of the surface-air interface to record satellite-based geolocations or occupation of the photic zone to collect light levels to estimate position. It is common among fishes to avoid the photic zone during daylight hours thus rendering this geolocation approach useless. In the HMMoce package, we leverage all tag-based data streams, like depth-temperature profiles, in conjunction with whatever traditional geolocation data (e.g. light) is available to calculate the most probable movements of the tagged animal. This is performed in a hidden Markov framework originally developed by Pedersen et al. 2008.

Author(s)

Maintainer: Camrin Braun <camrin.braun@gmail.com>

Authors:

- Benjamin Galuardi <drdrumfish@gmail.com>

Other contributors:

- Benjamin Jones (Contributed to earlier version of some of the download functions.) [contributor]
- Martin Pedersen (Developed an earlier version of some of the HMM framework and helper functions.) [contributor]

References

- Pedersen MW, Righton D, Thygesen UH, et al. (2008) Geolocation of North Sea cod (Gadus morhua) using hidden Markov models and behavioural switching. Can J Fish Aquat Sci 65:2367-2377.
- Pedersen MW, Patterson TA, Thygesen UH, Madsen H (2011) Estimating animal behavior and residency from movement data. Oikos 120:1281-1290.
- Woillez M, Fablet R, Ngo TT, et al. (2016) A HMM-based model to geolocate pelagic fish from high-resolution individual temperature and depth histories: European sea bass as a case study. Ecol Modell 321:10-22.

See Also

Useful links:

- http://www.camrinbraun.com/
- Report bugs at https://github.com/camrinbraun/HMMoce/issues

---

`likint3` *Calculate density function and integrate between limits*

Description

#' `likint3` calculates density function for a normal distribution and integrates between limits

Usage

```
likint3(w, wsd, minT, maxT)
```

Arguments

`w`	is an array of grid values like sea surface temperature
`wsd`	is an array containing the sd of the grid values, usually from `raster::focal`
`minT`	is an integer representing the lower limit of the tag-measured variable (e.g. SST)
`maxT`	is an integer representing the upper limit of the tag-measured variable

Value

an array of dim(w) that represents the likelihood of the tag-measured variable as compared to the input grid

Examples

```
# Dummy example environmental grid
env.grid <- matrix(seq(1,100,by=1), ncol=10)

# Generates likelihood of measurement
# as compared to environmental grid
likint3(env.grid, 2, 55, 70)
```

liksrss *Calculate density function and integrate between limits*

Description

#' liksrss calculates density function for a normal distribution and integrates between limits of SRSS times (dawn/dusk)

Usage

```
liksrss(obs, srss, srsd)
```

Arguments

obs	input light observation. See calc.srss
srss	is raster of possible SR or SS times within study area
srsd	is raster of SD of SRSS times in study area calculated with raster::focal

Value

an array of SRSS-based likelihoods

make.L *Combine individual source likelihoods*

Description

make.L combines individual likelihoods from various data sources (e.g. SST, OHC) to make overall combined likelihoods for each time point

Usage

```
make.L(L1, L2 = NULL, L3 = NULL, known.locs = NULL, L.mle.res,
  dateVec = NULL, locs.grid = NULL, iniloc = NULL, bathy = NULL,
  pdt = NULL)
```

Arguments

L1	a likelihood array
L2	a likelihood array
L3	a likelihood array
known.locs	is data frame of known locations containing named columns of date, lon, lat. Default is NULL.

L.mle.res	is a coarse resolution array of dim(L1) that speeds up the parameter estimation step later on
dateVec	is vector of dates from tag to pop-up date by day. Only required if known.locs is not NULL.
locs.grid	is output grid from setup.locs.grid. Only required if known.locs is not NULL.
iniloc	is matrix of tag and pop locations. Default is NULL because this should be taken care of elsewhere.
bathy	is bathymetry raster (likely from ETOPO1) as acquired by get.bath.data
pdt	is data frame output from read.wc(type='pdt')

Value

a list containing: L, the overall likelihood array and L.mle, a more coarse version of L used later for parameter estimation

Note

This function currently only supports the use of 3 input likelihood data sources. This will be expanded in the future based on user needs.

---

makePar makePar *gets parameters for subsequent filter/smoother*

Description

Function builds movement kernels for 2 different behavior states and calculates, if desired, switching probability using an expectation maximization routine

Usage

```
makePar(migr.spd, grid, L.arr, p.guess = c(0.7, 0.8), calcP = FALSE)
```

Arguments

migr.spd	is numeric input to calc.param
grid	is a grid output by resample.grid that corresponds to the extent and resolution of L.arr (below).
L.arr	is the likelihood array used for state switch probability calculation (see expmax). This is typically the L.mle array returned from make.L because it's typically more coarse (and thus faster) than the higher-resolution L array.
p.guess	is vector of length 2 indicating probability of staying in states 1 and 2, respectively
calcP	is logical indicating whether to use expmax to calculate state-switching probabilities

Value

list of parameters including movement kernels (K1, K2) and switch probability (P.final)

Examples

```
## Not run:
par0 <- makePar(migr.spd=2, grid=g.mle, L.arr=L.mle, p.guess=c(.9,.9), calcP=T)

## End(Not run)
```

`mask.L` *Mask L likelihood*

Description

`mask.L` masks L likelihood based on a certain extent from previous days filter prediction

Usage

```
mask.L(pred.t, L.t, lon, lat, par0, bound.thr = 0.05, minBounds = NULL)
```

Arguments

`pred.t`	is prediction at t-1 that is used for 1) the size of the mask and 2) multiplying with L for resulting post likelihood surface
`L.t`	is data based likelihood layer, L[t], for the time step of interest (usually a day)
`lon`	vector of longitude values corresponding to dims of the previous 2 layers
`lat`	vector of latitude values corresponding to dims of the previous 2 layers
`par0`	is vector of movement parameter values, likely output from `calc.param`.
`bound.thr`	is numeric indicating the percent threshold that is added and subtracted from the bounding box of the filter output from the previous day before masking. Default is .05 (5 percent).
`minBounds`	is size (in degrees) of the minimum bounding box around the previous days filter prediction that L data within that box will be included. Outside this box (centered on t-1 filter prediction), L will be masked out.

Details

L likelihood is masked based on a percent size (in addition to) the extent of the previous days prediction kernel. If that kernel is smaller than the migratory kernel being used, the migratory kernel size is defaulted to. User can also specify a minimum bound size.

Value

post matrix as a product of the input prediction and L values

meshgrid *Creates grid from matrices*

Description

Creates grid from matrices

Usage

```
meshgrid(x, y)
```

Arguments

x	vector, usually of longitude data
y	vector, usually of latitude data

Value

list of 2 matrices for lon/lat values

References

Pedersen MW, Patterson TA, Thygesen UH, Madsen H (2011) Estimating animal behavior and residency from movement data. Oikos 120:1281-1290. doi: 10.1111/j.1600-0706.2011.19044.x

Examples

```
## Not run:
x <- c(1, 2, 3)
y <- c(10, 9, 8)
meshgrid(x, y)

## End(Not run)
```

plotHMM *Plot track results of HMMoce*

Description

`plotHMM` uses HMM output via `calc.track` to make simple plots of calculated track and behavior state

Usage

```
plotHMM(distr, track, dateVec, ptt, known = NULL, resid = FALSE,
  behav.pts = F, save.plot = FALSE)
```

Arguments

distr	is output array from hmm.smoother
track	is output dataframe from calc.track
dateVec	is vector of dates from tag to pop-up location by day.
ptt	is unique ID for individual tag dataset
known	is 3 column data frame containing date, lat, lon of known movement track. This is only useful for comparing HMMoce results to known track collected by SPOT or GPS, for example. Default is NULL.
resid	is logical indicating whether you want to include a residual plot. This is not yet functional.
behav.pts	is logical indicating whether to plot points in the resulting map colored by behav state.
save.plot	is logical indicating whether you want the plot written to disk using pdf.

Value

NULL. A plot is rendered on screen or written to disk.

References

Pedersen MW, Patterson TA, Thygesen UH, Madsen H (2011) Estimating animal behavior and residency from movement data. Oikos 120:1281-1290. doi: 10.1111/j.1600-0706.2011.19044.x

---

plotRD *Plot RD results*

Description

plotRD uses HMM output via calc.track to calculate and plot residency distributions for each behavior state and combined data

Usage

```
plotRD(distr, track, ptt, known = NULL, g, xlims, ylims, makePlot = TRUE,
  save.plot = FALSE)
```

Arguments

distr	is output array from hmm.smoother
track	is output dataframe from calc.track
ptt	is individual identification number
known	is 3 column data frame containing date, lat, lon of known movement track. This is only useful for comparing HMMoce results to known track collected by SPOT or GPS, for example. Default is NULL.

g	grid from setup.grid
xlims	a vector of length 2 indicating longitude limits (-180 to 180).
ylims	a vector of length 2 indicating latitude limits
makePlot	is logical indicating whether to make a plot of the RD
save.plot	is logical indicating whether you want the plot written to disk using pdf.

Value

a list of one raster layer for the combined RD and one raster brick (2 layers) for the individual behavior RDs.

---

read.wc *Read and format tag data*

Description

read.wc reads and formats tag data output from Wildlife Computers Data Portal

Usage

```
read.wc(ptt, filename, tag, pop, type = "sst", verbose = FALSE)
```

Arguments

ptt	is individual ID number
filename	is path to the file where your data lives
tag	is POSIXct object of the tagging date
pop	is POSIXct object of the pop-up date
type	is character indicating which type of data to read. Choices are 'sst', 'pdt', 'light' corresponding to those data files output from WC Data Portal
verbose	is logical indicating whether a verbose output with more details on the loaded files is desired. Default is FALSE.

Value

a list containing: the data read as a data.frame and a date vector of unique dates in that data

Examples

```
# example data in the package
sstFile <- system.file("extdata", "141259-SST.csv", package = "HMMoce")
ptt <- 141259

# set temporal and spatial bounds
iniloc <- data.frame(matrix(c(13, 10, 2015, 41.3, -69.27, 10, 4, 2016, 40.251, -36.061),
 nrow = 2, ncol = 5, byrow = TRUE))
 names(iniloc) <- list('day','month','year','lat','lon')
 tag <- as.POSIXct(paste(iniloc[1,1], '/', iniloc[1,2], '/', iniloc[1,3], sep=''),
 format = '%d/%m/%Y', tz='UTC')
 pop <- as.POSIXct(paste(iniloc[2,1], '/', iniloc[2,2], '/', iniloc[2,3], sep=''),
 format = '%d/%m/%Y', tz='UTC')

# read and format the example data
tag.sst <- read.wc(ptt, sstFile, type = 'sst', tag=tag, pop=pop)
```

---

removePacific *Remove Pacific Ocean data from N. Atlantic analyses*

Description

removePacific removes Pacific Ocean from WOA and other forms of array-based data. This is a specialized function to address the issue when the Pacific side of Panama enters into the model bounding box of a North Atlantic analysis.

Usage

```
removePacific(dat, lat, lon)
```

Arguments

dat	is output from extract.woa
lat	is output from extract.woa
lon	is output from extract.woa

Value

dat is WOA data grid with Pacific removed only tested when area of interest is N Atlantic

Examples

```
## Not run:
woa.dir <- getwd()
woa <- extract.woa(woa.dir, bbox = c(-90, -30, -10, 30), 'quarter')
woa <- removePacific(woa, lat, lon)
image.plot(woa[,,1,1])

## End(Not run)
```

repmat *Repeat your matrix?*

Description

Repeat your matrix?

Usage

```
repmat(X, m, n)
```

Arguments

X	is matrix you want to repeat
m	is output row dimension
n	is output col dimension

Value

repeated matrix

References

Pedersen MW, Patterson TA, Thygesen UH, Madsen H (2011) Estimating animal behavior and residency from movement data. Oikos 120:1281-1290.

resample.grid *Resample likelihood rasters to common resolution/extent*

Description

Resample likelihood rasters to common resolution/extent

Usage

```
resample.grid(L.rasters, L.res, mle.res = 0.75, bound = NULL)
```

Arguments

L.rasters	list of individual likelihood rasters generated by calc functions
L.res	raster or raster brick indicating desired output resolution of all likelihood rasters.
mle.res	is desired resolution (in degrees) of the coarse output grid to be used in parameter estimation.
bound	is a raster extent object that can be passed in order to restrict the input raster likelihoods to a certain spatial domain. Default is null and thus uses the full extent of input rasters. For more information on the extent object see raster::extent

Value

a list of all resampled likelihood rasters and g, the common grid

---

resample.grid.par *Resample likelihood rasters to common resolution/extent*

Description

Resample likelihood rasters to common resolution/extent

Usage

```
resample.grid.par(L.rasters, L.resol, ncores = NULL)
```

Arguments

L.rasters	list of individual likelihood rasters generated by calc functions
L.resol	raster or raster brick indicating desired output resolution of all likelihood rasters.
ncores	is integer indicating number of cores used in this parallel computation. Defaults to using a detection function that chooses cores for you.

Value

a list of all resampled likelihood rasters and g, the common grid

Note

This function should probably only be used in special use cases. Otherwise, the non-parallel version resample.grid is typically faster.

---

setup.grid *Setup the discrete spatial grid for the HMM*

Description

Setup the discrete spatial grid for the HMM

Usage

```
setup.grid(locations, res)
```

Arguments

locations	dataframe of -Locations from WC psat tag
res	character indicating resolution of grid. 'hycom' is .08 to match hycom reanalysis res. 'quarter' and 'one' are .25 and 1 deg, respectively.

Value

a list

References

Pedersen MW, Patterson TA, Thygesen UH, Madsen H (2011) Estimating animal behavior and residency from movement data. Oikos 120:1281-1290. doi: 10.1111/j.1600-0706.2011.19044.x

---

setup.grid.raster *Setup the discrete spatial grid for the HMM*

Description

Setup the discrete spatial grid for the HMM

Usage

```
setup.grid.raster(grid.ras)
```

Arguments

grid.ras	is a raster for which a grid is desired

Value

a list

---

setup.locs.grid *Setup the discrete spatial grid for the HMM*

Description

`setup.locs.grid` sets up a discrete spatial grid for the HMM

If input limits is a data.frame, longitude limits become +/- 5 degrees and latitude becomes +/- 10 degrees. You may also input your own list of limits.

Usage

```
setup.locs.grid(limits, res = "quarter")
```

Arguments

limits	can take either a 'data.frame' of -Locations from WC psat tag or a pre-determined list of limits as `list(xmin, xmax, ymin, ymax)`
res	character indicating resolution of grid. 'hycom' is .08 to match hycom reanalysis res. 'quarter' and 'one' are .25 and 1 deg, respectively.

Value

a list of spatial bounds

References

Pedersen MW, Patterson TA, Thygesen UH, Madsen H (2011) Estimating animal behavior and residency from movement data. Oikos 120:1281-1290. doi: 10.1111/j.1600-0706.2011.19044.x

Examples

```
# SET SPATIAL LIMITS
sp.lim <- list(lonmin = -95, lonmax = -52, latmin = 10, latmax = 55)
locs.grid <- setup.locs.grid(sp.lim)
```

---

simplifyLocs *Simplifies locations to one per day*

Description

Simplifies locations to one per day

Usage

```
simplifyLocs(locations, loc.dts)
```

Arguments

locations	is data frame from -Locations file output from DAP/Tag Portal for WC tags and contains GPS, Argos, and GPE locations as applicable.
loc.dts	is vector of dates from locations input

Value

simplified (one per day) version of locations input

Index