Sea surface temp and known locations

Bindoff, A., Foo, D.

12 December 2017

# recommended install.packages("raster", repos = "https://cran.csiro.au") for latest version of `raster` package  
# devtools::install\_github("SWotherspoon/BAStag")  
  
library(SGAT)  
library(BAStag)  
library(raster)  
library(readr)  
library(TwilightFree)  
library(maptools)

For marine animals who traverse large distances across the latitudes, sea surface temperature (SST) data which are collected by most GLS tags can be used to improve location estimates. TwilightFree is able to use this information, but there are some common issues that users face which we aim to address in this tutorial.

This tutorial also demonstrates one of two methods for incorporating data about fixes at known locations, either from physical sightings, or known beaching, burrowing, or nesting locations and dates. These are sometimes evident from patterns in the light data, and can be confirmed by position estimates from an initial run that go close to the beach, burrow, or nest.

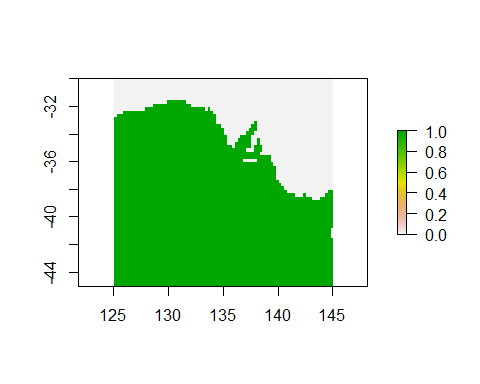
If you have *not* forked this repository from github, please download the SST data at <https://github.com/ABindoff/TwilightFree/blob/master/tutorials/sst_2014_2017.RData> (forking the repository avoids this step). This rasterStack covers mean weekly SST for the the years 2014-2017 and was obtained [from NOAA](https://www.esrl.noaa.gov/psd/repository/entry/show?entryid=12159560-ab82-48a1-b3e4-88ace20475cd). If you require data that is older or newer, download the file from NOAA and use sst <- stack("C:/Reynolds/sst.wkmean.1990-present.nc",varname="sst",quick=TRUE) to load it as a raster::rasterStack object.

load("../tutorials/sst\_2014\_2017.RData") ## `sst` download from https://github.com/ABindoff/TwilightFree/blob/master/tutorials/sst\_2014\_2017.RData

#### Define the extent of the grid

We know that this animal (a New Zealand fur seal) was tagged at Kangaroo Island, South Australia, and hasn't ranged far. makeGrid takes lon and lat extents, a cell size (in degrees), and optionally an argument to define a land/sea mask. pacific = T tells makeGrid that we wish to use Pacific-centered coordinates.

grid <- makeGrid(c(125, 145), c(-45,-30), cell.size = 1/4, pacific = T)  
plot(grid)



#### Load light and SST data

Light and SST data are typically stored in separate files (to save storage space on the tag) and need to be merged. *(The BAStag package offers tools that make this easy for BAS tags).*

# light and temp data are often in separate files  
d.lig <- read.csv("https://raw.githubusercontent.com/ABindoff/TwilightFree/master/tutorials/nzfs.lig.csv")  
head(d.lig)

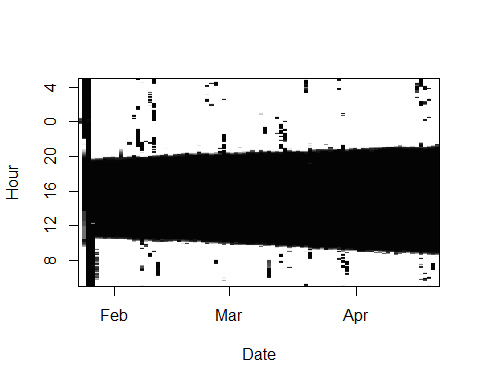
## Date Light  
## 1 23/01/2017 23:43 21.576  
## 2 23/01/2017 23:48 20.440  
## 3 23/01/2017 23:53 6.813  
## 4 23/01/2017 23:58 7.949  
## 5 24/01/2017 0:03 46.558  
## 6 24/01/2017 0:08 13.627

Note the format of the Date column. This needs to be converted to POSIXct.

# fix Date to %Y-%m-%d %H:%M:%S format  
d.lig$Date <- as.POSIXct(strptime(as.character(d.lig$Date), "%d/%m/%Y %H:%M", tz="GMT"))  
  
# read temp data  
d.tem <- read.csv("https://raw.githubusercontent.com/ABindoff/TwilightFree/master/tutorials/nzfs.tem.csv")  
d.tem$Date <- as.POSIXct(strptime(as.character(d.tem$Date), "%d/%m/%Y %H:%M", tz="GMT"))  
  
# align temp observations with light observations  
d.lig$Temp[d.lig$Date %in% d.tem$Date] <- d.tem$Temp[d.tem$Date %in% d.lig$Date]  
  
# check the aligned data and view image  
d.lig[c(3456:3460,3500:3504),]

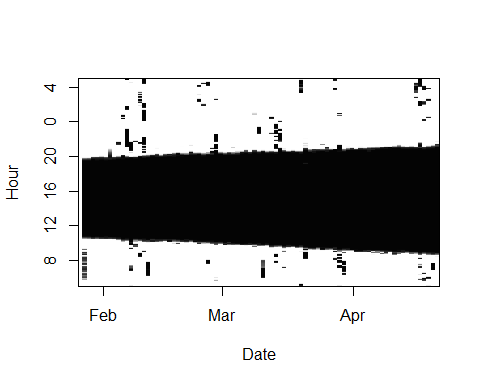
## Date Light Temp  
## 3456 2017-02-04 23:38:00 13954.84 19.1  
## 3457 2017-02-04 23:43:00 12792.03 NA  
## 3458 2017-02-04 23:48:00 41862.24 NA  
## 3459 2017-02-04 23:53:00 41862.24 NA  
## 3460 2017-02-04 23:58:00 46513.47 NA  
## 3500 2017-02-05 03:18:00 30234.15 NA  
## 3501 2017-02-05 03:23:00 34885.39 NA  
## 3502 2017-02-05 03:28:00 34885.39 NA  
## 3503 2017-02-05 03:33:00 32559.77 NA  
## 3504 2017-02-05 03:38:00 17443.26 19.4

lightImage(d.lig, offset = 5, zlim = c(0, 64))



Note the noise at the beginning of the light data, this is possibly because the tag was switched on and left in a tent or bag prior to deployment. We need to trim appropriately.

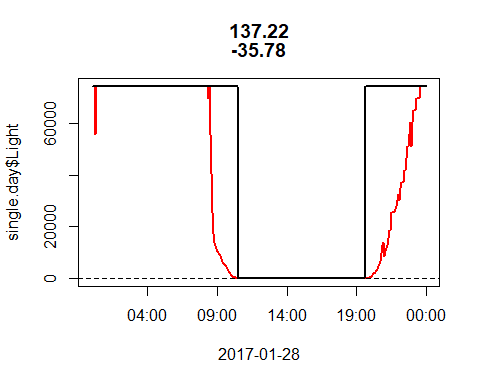
# chop the calibration and transit periods off the ends  
d.lig <- subset(d.lig,Date >= as.POSIXct("2017-01-27 00:00",tz = "GMT") &  
 Date < as.POSIXct("2017-04-21 00:00",tz = "GMT"))  
lightImage(d.lig, offset = 5, zlim = c(0,64))



#### Threshold and solar zenith angle

Use calibrate to determine the threshold and solar zenith angle used in the model.

# find optimal threshold and solar zenith angles for tag using `calibrate`  
zen <- 96  
day <- as.POSIXct("2017-01-28 00:00", "GMT")  
thresh <- calibrate(d.lig, day, 137.22, -35.78, zen) \*1.01



## [1] "max light in night window: 12.491 assuming a solar zenith angle of: 96"

#### Fixes

We have a small list of dates where the animal returned to the colony. This includes the deployment and retrieval dates. Normally this might be a .csv or .txt file, we just need Date, Lon, and Lat columns in a data frame. We pass these to the fixd argument in TwilightFree. We also pass sst, zenith, threshold, and the hyperparameters relating to sensor obscuration (alpha) and movement (beta). We may not specify deployed.at or retrieved.at parameters if we supply fixd.

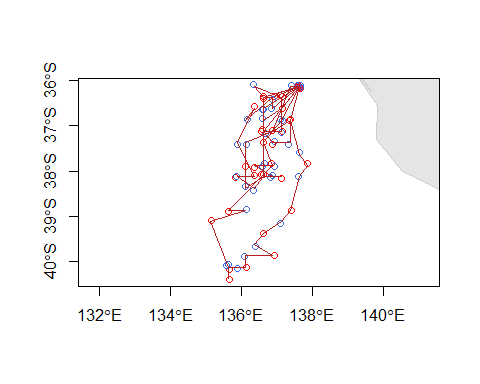
# dates where animal returned to the colony were recorded in field notes  
# and recorded in this spreadsheet (but the Date column needs formatting)  
sightings <- data.frame(Date = c("2017-01-28", "2017-02-11", "2017-02-28", "2017-03-15", "2017-03-20", "2017-04-19"),  
 Lon = 137.5,  
 Lat = -36.1)  
  
model <- TwilightFree(d.lig,  
 alpha=c(1, 1/5),  
 beta=c(1, 1/4),  
 zenith = zen, threshold = thresh,  
 fixd = sightings, # these are the colony locations and dates  
 sst = sst) # this is the sst raster stack

Pass the TwilightFree model object to SGAT::essie which calculates the posterior with a forward-backward algorithm.

# fit the model using the grid from `makeGrid`  
fit <- SGAT::essie(model,grid,epsilon1=1.0E-4, epsilon2 = 1E-4)

trip() will return the locations from the fitted essie object, it's a good idea to save these to a file. TwilightFree offers a basic plotting function drawTracks which will plot the track, and optionally another track (say, from a GPS tag). See ? drawTracks for documentation.

# plot the result  
locs <- trip(fit)  
drawTracks(locs, pacific = T)



An optional step is to smooth the track using a state-space model such as bsam. You will need to install jags (on your computer, not in R) in order to use bsam, so this chunk is not evaluated by default (set eval = T and install jags and bsam to knit it).

The "trick" is to tell bsam that the GLS locations estimated using TwilightFree method are Argos locations that need smoothing.

# smooth using a state space model  
#install.packages("bsam")  
library(bsam)  
locs$Argos\_loc\_class <- "G" ## set all to class "G" (trick bsam into thinking it's Argos data)  
locs$lonerr <- 1 ##error in deg  
locs$laterr <- 1  
locs$gmt <- as.POSIXct(paste(as.character(locs$Date), "12:00:00", sep=" "), tz="GMT")  
locs$ref <- 1  
  
d <- locs[, c("ref", "gmt", "Argos\_loc\_class","Lon", "Lat", "lonerr", "laterr")]  
colnames(d) <- c("id", "date","lc", "lon", "lat" , "lonerr", "laterr")  
  
  
fit <- fit\_ssm(d, model = "DCRW", tstep = 1, adapt = 5000, samples = 5000,  
 thin = 5, span = 0.2)  
  
# diag\_ssm(fit)  
# plot\_fit(fit)  
result <- get\_summary(fit)  
result$Lon <- result$lon  
result$Lat <- result$lat  
drawTracks(result, pacific = T)