Using trip - an R package for summarizing animal track data

Michael D. Sumner

August 10, 2006

Contents

1 Introduction

The `trip` package provides functions for summarizing animal track data. It is written using the classes provided by the R spatial package `sp`. The basic data component used is a data frame, with the minimal fields in each record of x and y coordinates, date-times and ID. This document demonstrates some examples using the package for importing data and dealing with common problems, for "filtering" and "gridding" track data, and exporting results.

`trip` replaces the experimental package `timeTrack` version 1.1-6, which was more limited in scope.

The package was written with broader applications in mind, including light level geolocation and Bayesian statistical methods for estimating location uncertainty. The hope is that capabilities will be simply added without disrupting existing functions. However, as usual there is no guarantee that things won't change - it's always important to know what version you have and how it works. For now, `trip` may be obtained from me directly.

General support for R is always to be found at `http://www.r-project.org/`, but I must take all the blame for the package `trip`. Please contact me directly for help and queries. For general `sp` and other queries on spatial data in R, there is the R-Sig-GEO mailing list.

Objects of class `trip` are a simple extension to the `sp` class `SpatialPointsDataFrame`. This is done with a new class `TimeOrderedRecords`, which is merely a place-holder for the names of the date-time and ID columns for trips. The `trip` package is loaded by

```
> library(trip)
```

```
Loading required package: sp
Loading required package: rgdal
Loading required package: abind
Loading required package: pixmap
Geospatial Data Abstraction Library extensions to R successfully loaded
```

2 Getting started: problem-free data from Argos DAT files

Argos (DAT) files can be read directly using `readArgos`, but any table data of coordinates and times may be used.

For data from Argos DAT files that require no further quality control, the function `readArgos` will return a `trip` object.

```
> argosfiles <- list.files(path = "C:/temp/blackBrowed/", pattern = ".dat",
+     full.names = TRUE)
> tr <- readArgos(argosfiles[1:3])
```

```
Adjusting duplicate times
.....
      ptt                 gmt class row.number
144 14257 2001-12-11 19:36:24     0        144
145 14257 2001-12-11 20:20:30     0        145
146 14257 2001-12-11 20:20:30     A        146
147 14257 2001-12-11 22:01:04     1        147
      ptt                 gmt class row.number
178 14257 2001-12-13 09:55:52     1        178
179 14257 2001-12-13 11:34:27     2        179
180 14257 2001-12-13 11:34:27     A        180
181 14257 2001-12-13 13:16:24     1        181
       ptt                 gmt class row.number
1010 14403 2001-12-02 10:35:28     A        455
1110 14403 2001-12-02 10:41:08     2        456
1210 14403 2001-12-02 10:41:08     B        457
1310 14403 2001-12-02 12:16:16     1        458
       ptt                 gmt class row.number
2010 14403 2001-12-02 16:33:41     0        465
2110 14403 2001-12-02 17:14:08     1        466
2210 14403 2001-12-02 17:14:08     0        467
2310 14403 2001-12-02 18:12:59     2        468
       ptt                 gmt class row.number
1452 14418 2001-12-06 14:45:23     0       1069
1462 14418 2001-12-06 14:57:38     1       1070
```

```
1472 14418 2001-12-06 14:57:38     B       1071
1482 14418 2001-12-06 16:23:33     1       1072
       ptt                  gmt class row.number
4122 14418 2001-12-16 18:28:03     1       1336
4132 14418 2001-12-16 19:21:52     0       1337
4142 14418 2001-12-16 19:21:52     0       1338
4152 14418 2001-12-16 20:10:44     0       1339

  Adjusted records now:

      ptt                  gmt class row.number
144 14257 2001-12-11 19:36:24     0        144
145 14257 2001-12-11 20:20:30     0        145
146 14257 2001-12-11 20:20:31     A        146
147 14257 2001-12-11 22:01:04     1        147
      ptt                  gmt class row.number
178 14257 2001-12-13 09:55:52     1        178
179 14257 2001-12-13 11:34:27     2        179
180 14257 2001-12-13 11:34:28     A        180
181 14257 2001-12-13 13:16:24     1        181
       ptt                  gmt class row.number
1010 14403 2001-12-02 10:35:28     A        455
1110 14403 2001-12-02 10:41:08     2        456
1210 14403 2001-12-02 10:41:09     B        457
1310 14403 2001-12-02 12:16:16     1        458
       ptt                  gmt class row.number
2010 14403 2001-12-02 16:33:41     0        465
2110 14403 2001-12-02 17:14:08     1        466
2210 14403 2001-12-02 17:14:09     0        467
2310 14403 2001-12-02 18:12:59     2        468
       ptt                  gmt class row.number
1452 14418 2001-12-06 14:45:23     0       1069
1462 14418 2001-12-06 14:57:38     1       1070
1472 14418 2001-12-06 14:57:39     B       1071
1482 14418 2001-12-06 16:23:33     1       1072
       ptt                  gmt class row.number
4122 14418 2001-12-16 18:28:03     1       1336
4132 14418 2001-12-16 19:21:52     0       1337
4142 14418 2001-12-16 19:21:53     0       1338
4152 14418 2001-12-16 20:10:44     0       1339



 Data fully validated: returning object of class  trip

> summary(tr)
```

```
Object of class trip
  tripID ("ptt") No.Records   startTime ("gmt")     endTime ("gmt")
1          14257        445 2001-12-06 01:35:31 2001-12-27 04:40:19
2          14403        479 2001-12-02 04:03:06 2001-12-18 20:16:06
3          14418        684 2001-12-02 05:46:51 2001-12-27 06:18:30

Derived from Spatial data:

Object of class SpatialPointsDataFrame
Coordinates:
              min     max
longitude 147.872 189.025
latitude  -61.207 -37.800
Is projected: FALSE
proj4string : [ +proj=longlat +ellps=WGS84]
Number of points: 1608
Data attributes:
    prognum          ptt            nlines          nsensor  satname class
 Min.   :1807   Min.   :14257   Min.   : 2.000   Min.   :4   D:245   Z:  0
 1st Qu.:1807   1st Qu.:14257   1st Qu.: 4.000   1st Qu.:4   H:316   B:234
 Median :1807   Median :14403   Median : 6.000   Median :4   J:310   A:202
 Mean   :1807   Mean   :14369   Mean   : 6.342   Mean   :4   K:358   0:669
 3rd Qu.:1807   3rd Qu.:14418   3rd Qu.: 8.000   3rd Qu.:4   L:379   1:346
 Max.   :1807   Max.   :14418   Max.   :14.000   Max.   :4           2:135
                                                                     3: 22
         date            time         altitude   transfreq
 2001-12-09:  89   05:44:16:   2   Min.   :0   Min.   :401653551
 2001-12-10:  87   06:37:59:   2   1st Qu.:0   1st Qu.:401653710
 2001-12-06:  84   07:28:45:   2   Median :0   Median :401653830
 2001-12-11:  82   08:38:14:   2   Mean   :0   Mean   :401653849
 2001-12-17:  77   11:34:27:   2   3rd Qu.:0   3rd Qu.:401653970
 2001-12-07:  75   17:20:03:   2   Max.   :0   Max.   :401654168
 (Other)   :1114   (Other) :1596
      gmt
 Min.   :2001-12-02 04:03:06
 1st Qu.:2001-12-07 20:19:05
 Median :2001-12-12 20:09:02
 Mean   :2001-12-13 13:20:02
 3rd Qu.:2001-12-18 13:46:50
 Max.   :2001-12-27 06:18:30
```

(These data were provided by the DPIWE Macquarie Island Albatross Project, [**?**]). We import only three of the available Argos files for now.

In Argos DAT files the fields `longitude` and `latitude` contain the spatial coordinates (these have been extracted from the other data in the `SpatialPointsDataFrame` in the usual way), `date` and `time` the temporal information (these have been combined into an R POSIXct vector/column called gmt), and ptt is the ID for individual instruments that is used as the trip ID. readArgos will perform some sensible quality control corrections by default. The output in this example is a report on which records contained duplicate times, which are modified by one second. The summary command returns a listing of the individual trips, their ID, start and end times, and number of locations. The remaining data a summarized in the usual way for a SpatialPointsDataFrame.

2.1 Filtering for unlikely speeds

The trip data are of Black-Browed albatross from Macquarie Island. These animals can fly up to 100 km/hr and so we have a simplistic means of quality control by removing any locations that imply unrealistic motion. We create a "filter" (by applying a very strict constraint on speed for illustration) and add this logical column to our data frame. The filtering algorithm is that of [**?**].

```
> tr$ok <- speedfilter(tr, max.speed = 20)
> summary(tr)

Object of class trip
  tripID ("ptt") No.Records   startTime ("gmt")     endTime ("gmt")
1          14257        445 2001-12-06 01:35:31 2001-12-27 04:40:19
2          14403        479 2001-12-02 04:03:06 2001-12-18 20:16:06
3          14418        684 2001-12-02 05:46:51 2001-12-27 06:18:30

Derived from Spatial data:

Object of class SpatialPointsDataFrame
Coordinates:
              min     max
longitude 147.872 189.025
latitude  -61.207 -37.800
Is projected: FALSE
proj4string : [ +proj=longlat +ellps=WGS84]
Number of points: 1608
Data attributes:
    prognum          ptt            nlines          nsensor  satname class
 Min.   :1807   Min.   :14257   Min.   : 2.000   Min.   :4   D:245   Z:  0
 1st Qu.:1807   1st Qu.:14257   1st Qu.: 4.000   1st Qu.:4   H:316   B:234
 Median :1807   Median :14403   Median : 6.000   Median :4   J:310   A:202
 Mean   :1807   Mean   :14369   Mean   : 6.342   Mean   :4   K:358   0:669
 3rd Qu.:1807   3rd Qu.:14418   3rd Qu.: 8.000   3rd Qu.:4   L:379   1:346
 Max.   :1807   Max.   :14418   Max.   :14.000   Max.   :4           2:135
                                                                     3: 22
         date            time         altitude   transfreq
 2001-12-09:  89   05:44:16:   2   Min.   :0   Min.   :401653551
 2001-12-10:  87   06:37:59:   2   1st Qu.:0   1st Qu.:401653710
 2001-12-06:  84   07:28:45:   2   Median :0   Median :401653830
 2001-12-11:  82   08:38:14:   2   Mean   :0   Mean   :401653849
 2001-12-17:  77   11:34:27:   2   3rd Qu.:0   3rd Qu.:401653970
 2001-12-07:  75   17:20:03:   2   Max.   :0   Max.   :401654168
 (Other)   :1114   (Other) :1596
      gmt                            ok
 Min.   :2001-12-02 04:03:06   Mode :logical
 1st Qu.:2001-12-07 20:19:05   FALSE:4
 Median :2001-12-12 20:09:02   TRUE :1604
 Mean   :2001-12-13 13:20:02
 3rd Qu.:2001-12-18 13:46:50
 Max.   :2001-12-27 06:18:30
```

We can see by summary that a number of locations may be excluded using our new "ok" column. Although our speed filter has not removed many locations, we can also subset based on other data. This time we will choose a minimum Argos location quality "class". We plot the raw data (using `sp`'s default plot for a `SpatialPointsDataFrame`), and then add lines from only the filtered data (coloured using a supplied `trip` method). The plot is shown in figure 1.

```
> plot(tr, axes = TRUE)
> lines(tr[tr$ok & tr$class > "A", ])
```

2.2 Creating a map of time spent

Assuming that our filtered locations give us realistic information about position for the animal, and that motion between these positions is constant and straight, we can easily create a map of time spent. The choice of grid cell size might reflect our confidence in the accuracy of the location data, we might require a specific grain for comparison with another dataset, or we are simply interested in creating a pretty picture of animal behaviour using time per unit area as a proxy for foraging effort.

We use the function `tripGrid` with the subset of the trip object accepted by the speed filter to create a grid of time spent.
tripGrid will interpolate between positions based on a specified time duration, here we use one hour. A shorter period will result in a closer approximation to the total time spent, but will take longer to complete. This method is similar to that published by [**?**].

```
> trg <- tripGrid(tr[tr$ok, ], dur = 3600)

Using method  countGrid

lost seconds =  -1991433  out of a total  1507.824   hours
```

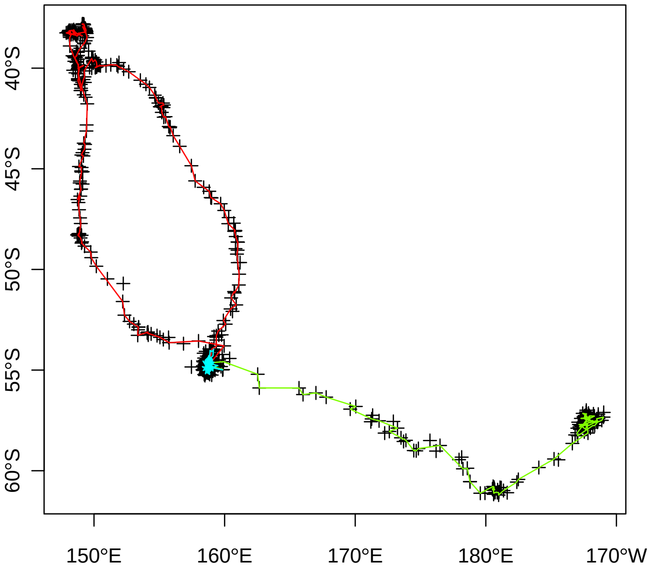

Figure 1: plot of `trip` as points, with lines coloured for each separate trip event

By default, tripGrid will provide a grid with dimensions 100x100 cells. To specify different size we can use the (trip) function makeGridTopology, to define a grid topology from the trip extents. This time, we subset the trip object using both the speed filter and a particular Argos quality ■class■. The first example shows the creation of a grid topology with dimensions of 50x50, then we create another using cellsize (assumed to be in kilometres for longlat or unspecified coordinate system).

```
> trA <- tr[tr$ok & tr$class > "A", ]
> gt <- makeGridTopology(trA, cells.dim = c(50, 50))
> gt
```

```
                        max     max.1
cellcentre.offset 146.87200 -62.20700
cellsize            0.86272   0.50814
cells.dim          50.00000  50.00000
```

```
> gt <- makeGridTopology(trA, cellsize = c(35, 20))
> trg <- tripGrid(trA, grid = gt, dur = 3600)
```

```
Using method  countGrid

lost seconds =  -1131520  out of a total  1502.689   hours
```