m2b tutorial

Laurent Dubroca, Andréa Thiebault

2017-03-10

Introduction

Animal behaviour, including social interactions, are fundamental to the field of ecology. Whereas the direct observation of animal behaviour is often limited due to logistical constraints, collection of movement data have been greatly facilitated through the development of bio-logging. Animal movement data obtained through tracking instrumentation may potentially constitute a relevant proxy to infer animal behaviour. This is, however, based on the premise that a range of movement patterns can be linked to specific behaviours.

Statistical learning constitutes a number of methods that can be used to assess the link between given variables from a fully informed training dataset and then predict the values on a non-informed variable. We chose the random forest algorithm for its capacity to deal with imbalanced data (particularly relevant for behavioural data), its high prediction accuracy and its ease of implementationi (Breiman (2001), Chen, Liaw, and Breiman (2004)). The strength of random forest partly relies in its ability to handle a very large number of variables. Hence, our methodology is based on the derivation of multiple predictor variables from the movement data over various temporal scales, in order to capture as much information as possible on the changes and variations of movement.

In this tutorial, the link between behaviour and movement parameters is build using a random forest model based on a seabird track with behavioural observation made by a video camera deployed on the individual. A new class named `xytb` is implemented in order to provide to the user an object where data, derived information, modelling and results are included in a single object.

A new class

`xytb` is a S4 class built to give the whole information related to a track (the track itself, the differentiation of the track parameters, the observed behaviour, the model and the prediction) in a single object. `xyt` stands for the track information, and `b` for the behaviour. 8 slots contains these information, and different methods are available to build the different part of the object. The rationale behind this object is the possibility to keep everything (data, model and prediction) in the same container, and by extension the possibility (1) to keep track of how precisely the prediction were made and (2) to exchange the analyses with different users easily.

Methods and function overview

The different analytical steps are summarized in 4 main functions and methods.

`xytb`

`xytb` is a class, but also a method. 4 distincts signature help the user to load tracks and behavioural information in the object. During this step, derivated information of the track are computed and stored in the slot `dxyt` and `befdxyt`. The parameter `winsize` and `idquant` let the user to specify the windows on which to compute statistical operator, and some of them (the quantiles mainly).

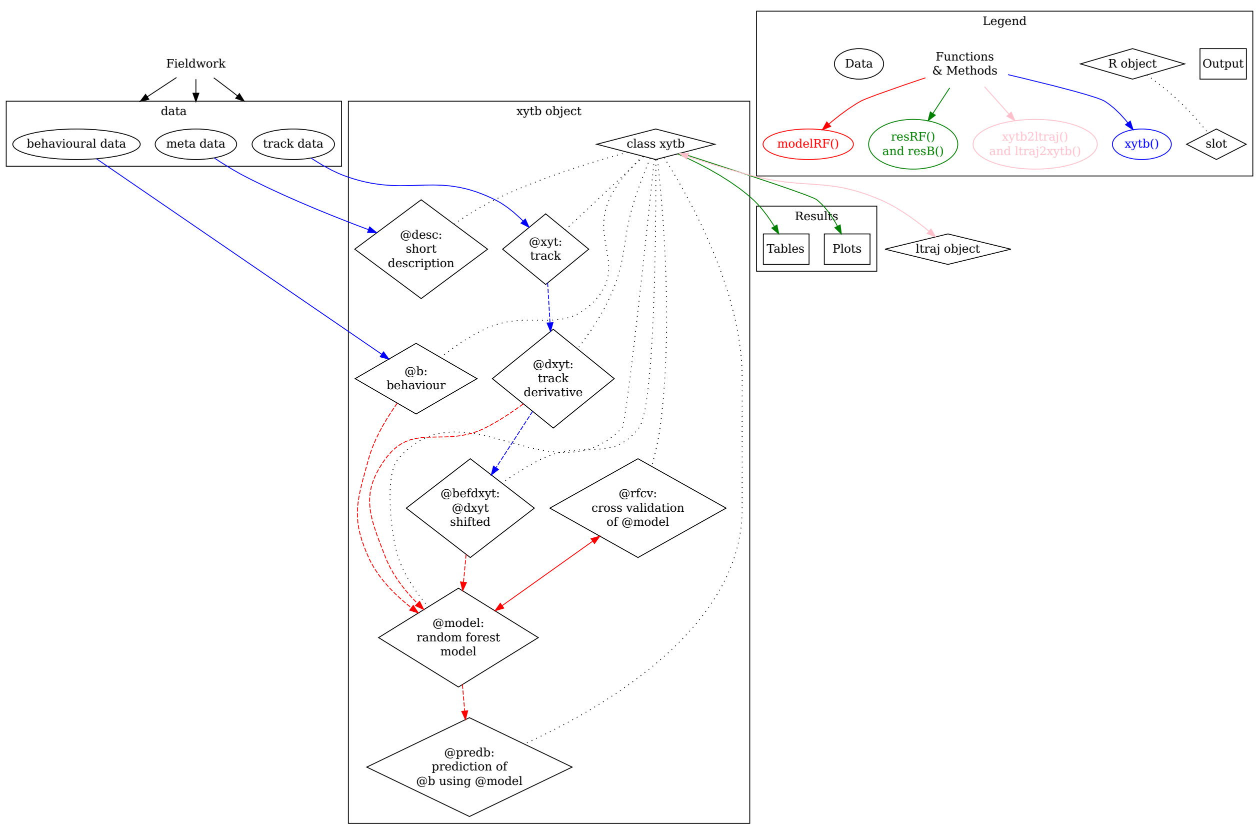


Figure 1: Schematic of the work flow from fieldwork to the results. The legend gives the representation of the different objects. The arrows between the boxes represent the use of the package functions and methods: in blue the computation of the data, in red the modelling, in green the outputs and in pink the export to the ltraj format (adehabitat package). The dotted lines between the xytb class and the R objects describe the slot of the class.

modelRF

This function compute a random forest model to predict the behavioural observation against the movement information.

resRF and resB

These two functions compute and plot the results related to the model (`resRF`) and the behaviours (prediction vs observation, `resB`).

xytb2ltraj and ltraj2xytb

These functions import or export an `xytb` object to the `ltraj` class. This class is linked to the `adehabitatLT` package and the numerous function dedicated to trajectories analyses (see Calenge, Dray, and Royer-Carenzi (2008)).

An example

Data

The data frame `track_CAGA_005` contains the information the track and the behavioural observation made on a gannet. Behavioural state at `-1` are unobserved behaviours. This parameters has to be passed to the functions during the analysis (see functions' help).

```
library(m2b)
str(track_CAGA_005)
```

```
## 'data.frame':    3597 obs. of  5 variables:
##  $ x : num  26.3 26.3 26.3 26.3 26.3 ...
##  $ y : num  -33.8 -33.8 -33.8 -33.8 -33.8 ...
##  $ t : POSIXct, format: "2010-12-11 08:08:00" "2010-12-11 08:08:13" ...
##  $ b : chr  "3" "3" "3" "3" ...
##  $ id: chr  "CAGA_005" "CAGA_005" "CAGA_005" "CAGA_005" ...
```

Different methods are available to build a `xytb` object. Here, the track and behavioural information are taken from the data.frame, and derived information related to the track are computed in the same time. The derived information are computed on the moving windows of 3, 5, 7, 9, 11, 13 and 15 locations (the `winsize` parameters), and with the standard statistical operators (mean, standard deviation and median absolute deviation) available in the methods, the quantile at 0, 25, 50, 75 and 100% are added (the `idquant` parameter). Then these values are shifted back in time at 5, 10 and 15 points backward (the `move` parameter), in order to build a dataset of predictor back in time. This is useful if the user is interested to investigate the link of the observed behaviour with previous movement events. The rationale behind this operation is based on the fact that some change in movement can be triggered by behavioural observation made afterward by the scientist but sooner by the animal.

```
library(m2b)
str(track_CAGA_005)
```

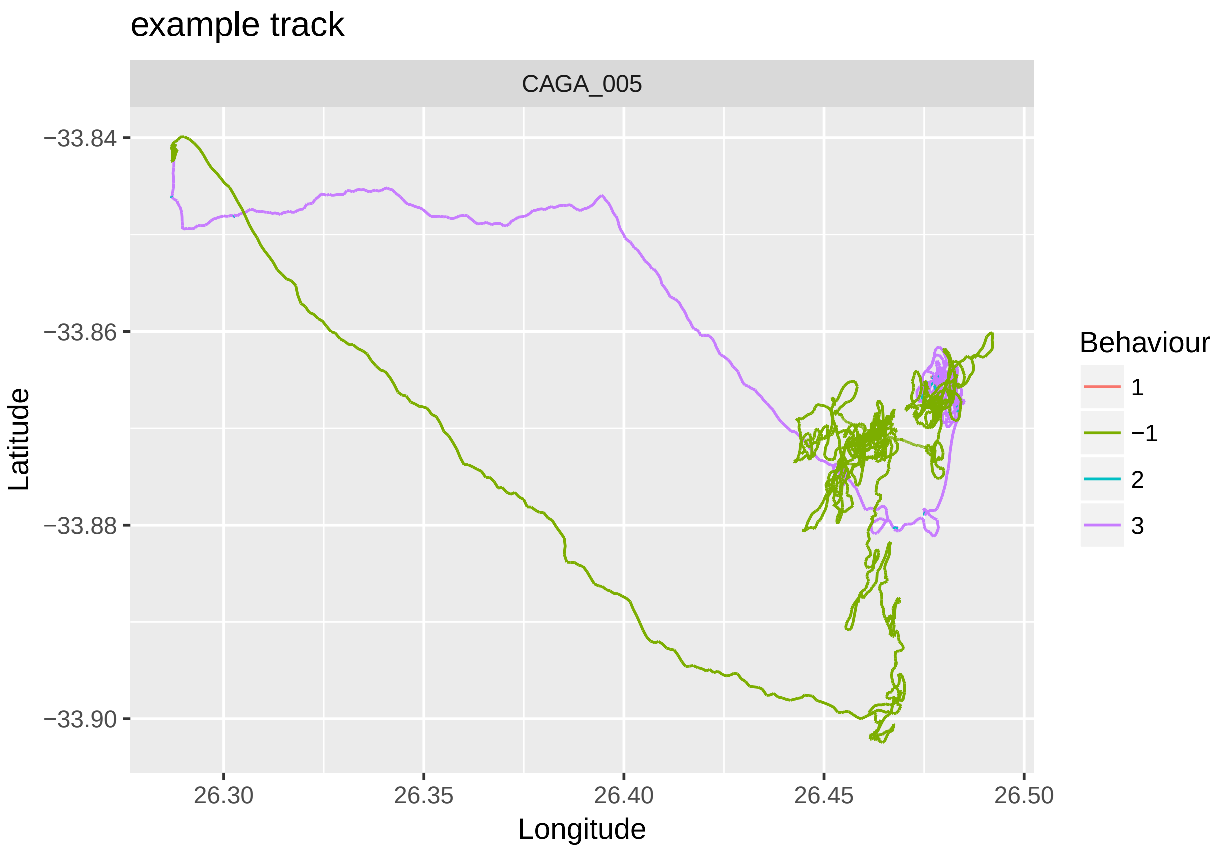
```
## 'data.frame':    3597 obs. of  5 variables:
##  $ x : num  26.3 26.3 26.3 26.3 26.3 ...
##  $ y : num  -33.8 -33.8 -33.8 -33.8 -33.8 ...
##  $ t : POSIXct, format: "2010-12-11 08:08:00" "2010-12-11 08:08:13" ...
##  $ b : chr  "3" "3" "3" "3" ...
##  $ id: chr  "CAGA_005" "CAGA_005" "CAGA_005" "CAGA_005" ...
```

```
#convert to xybt object with computation of windows operators and some quantiles
xytb<-xytb(track_CAGA_005,desc="example track",
     winsize=seq(3,15,2),idquant=seq(0,1,.25),move=c(5,10,15))
```

```
## [1] "Compute 169 indicators on 7 moving windows"
## [1] "Compute indicators on 3 points"
## [1] "Done"
## [1] "Compute indicators on 5 points"
## [1] "Done"
## [1] "Compute indicators on 7 points"
## [1] "Done"
## [1] "Compute indicators on 9 points"
## [1] "Done"
## [1] "Compute indicators on 11 points"
## [1] "Done"
## [1] "Compute indicators on 13 points"
## [1] "Done"
## [1] "Compute indicators on 15 points"
## [1] "Done"
## [1] "shift value backward"
## [1] "shift backward 5"
## [1] "shift backward 10"
## [1] "shift backward 15"
## [1] "Done"
```

```
#a simple plot method
plot(xytb)
```

```
## Warning in plyr::split_indices(scale_id, n): '.Random.seed' is not an
## integer vector but of type 'NULL', so ignored
```



Environnement

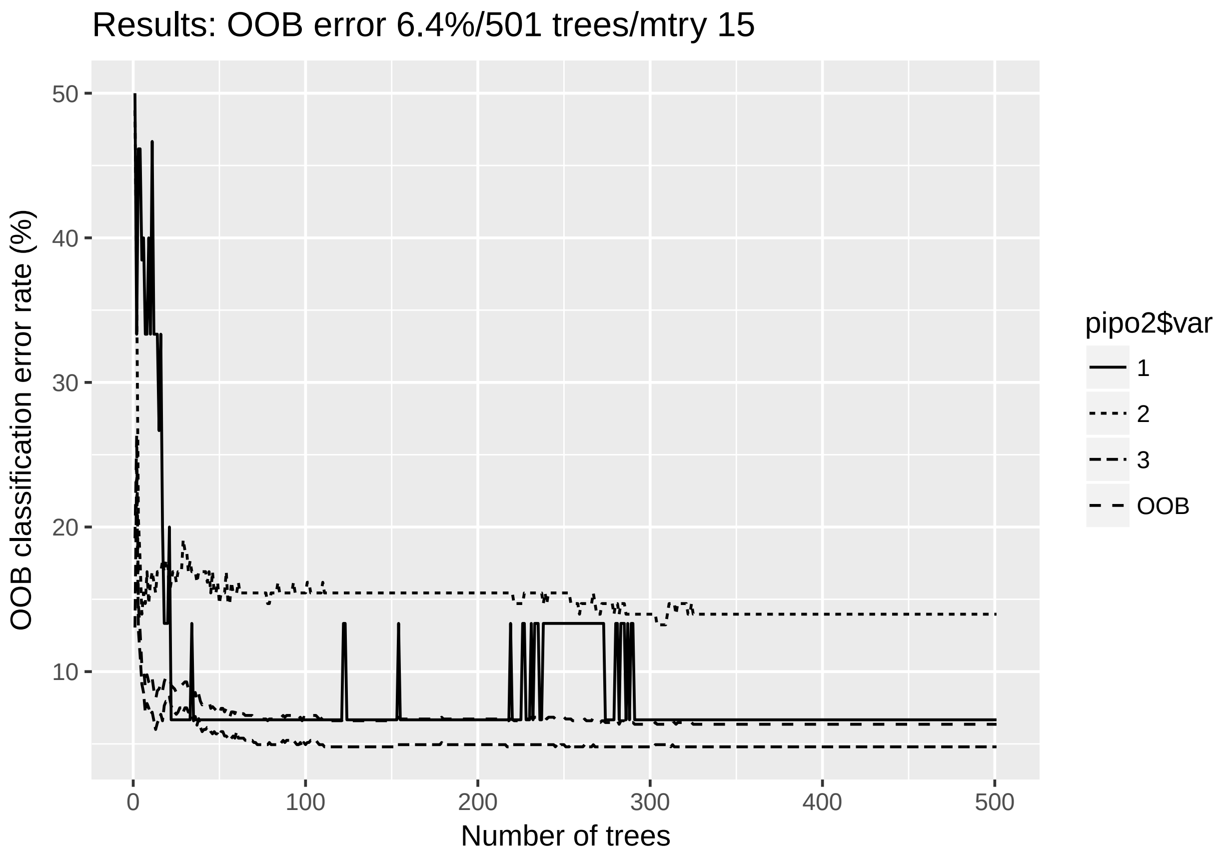
Modelling

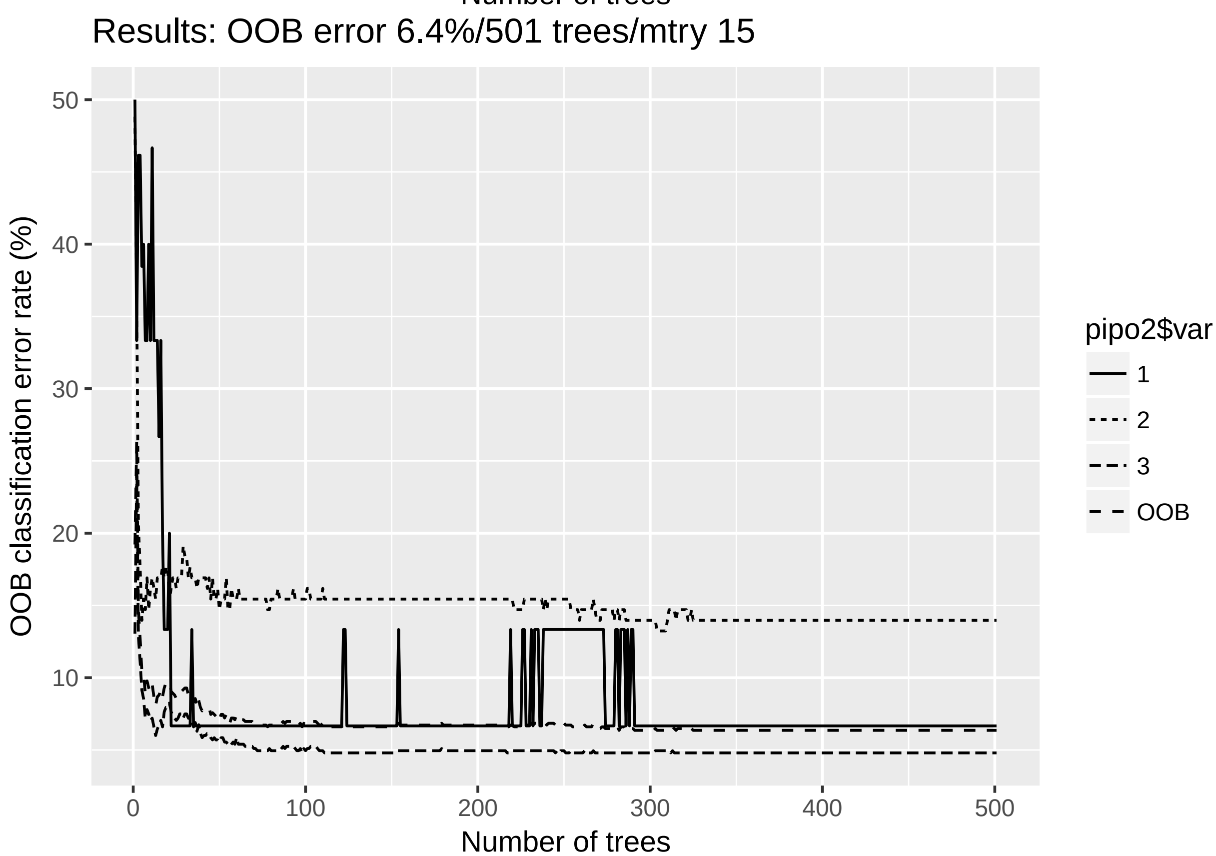
To build a random forest predicting the behavioural states based on the movement information, the function `modelRF` is used. It's a simple wrapper calling the `randomForest` function of the `randomForest` package, using the behavioural observation as response, and movement information as predictors. Some diagnosis are available to check the results using the `resRF` function, but the function `extractRF` made the model available in the `randomForest` format, and let the use of the other function of the `randomForest` package.

```
#a model (the function modelRF update the model inside the xytb object)
xytb<-modelRF(xytb,type="actual",ntree=501,mtry=15)
```

```
## [1] "removing lines with NA values"
## [1] "removing colinearity among predictors"
## [1] "v,dist,thetarel keeped"
## [1] "vmean_w3,distmean_w3,vmean_w5,distmean_w5,vmean_w7,thetarelmean_w7,distmean_w7,vsd_w7,distsd_w7
## [1] "removing near zero variance predictors"
```

```
resRF(xytb)
```

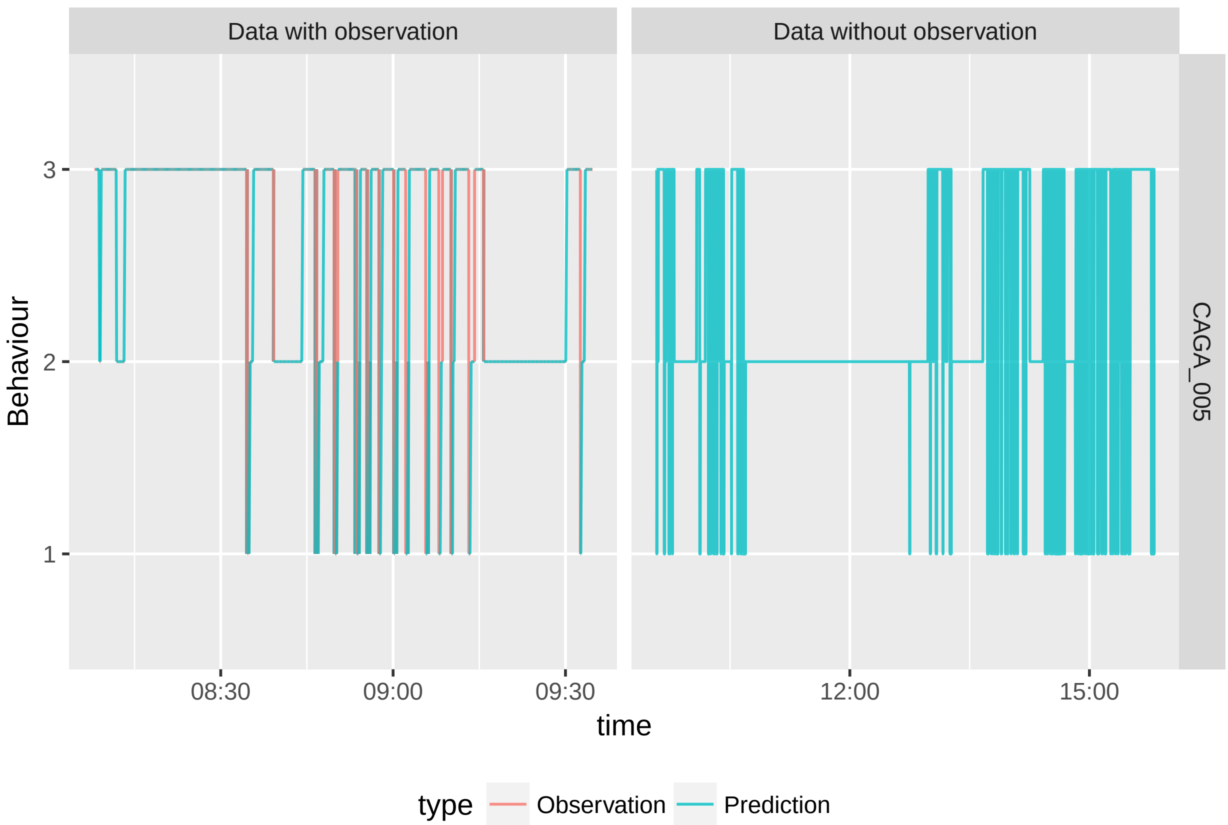

Results: OOB error 6.4%/501 trees/mtry 15
OOB classification error rate (%)
50
40
30
20
10
0
100
200
300
400
500
Number of trees
pipo2$var
1
2
3
OOB


Results: OOB error 6.4%/501 trees/mtry 15
OOB classification error rate (%)
50
40
30
20
10
0
100
200
300
400
500
Number of trees
pipo2$var
1
2
3
OOB

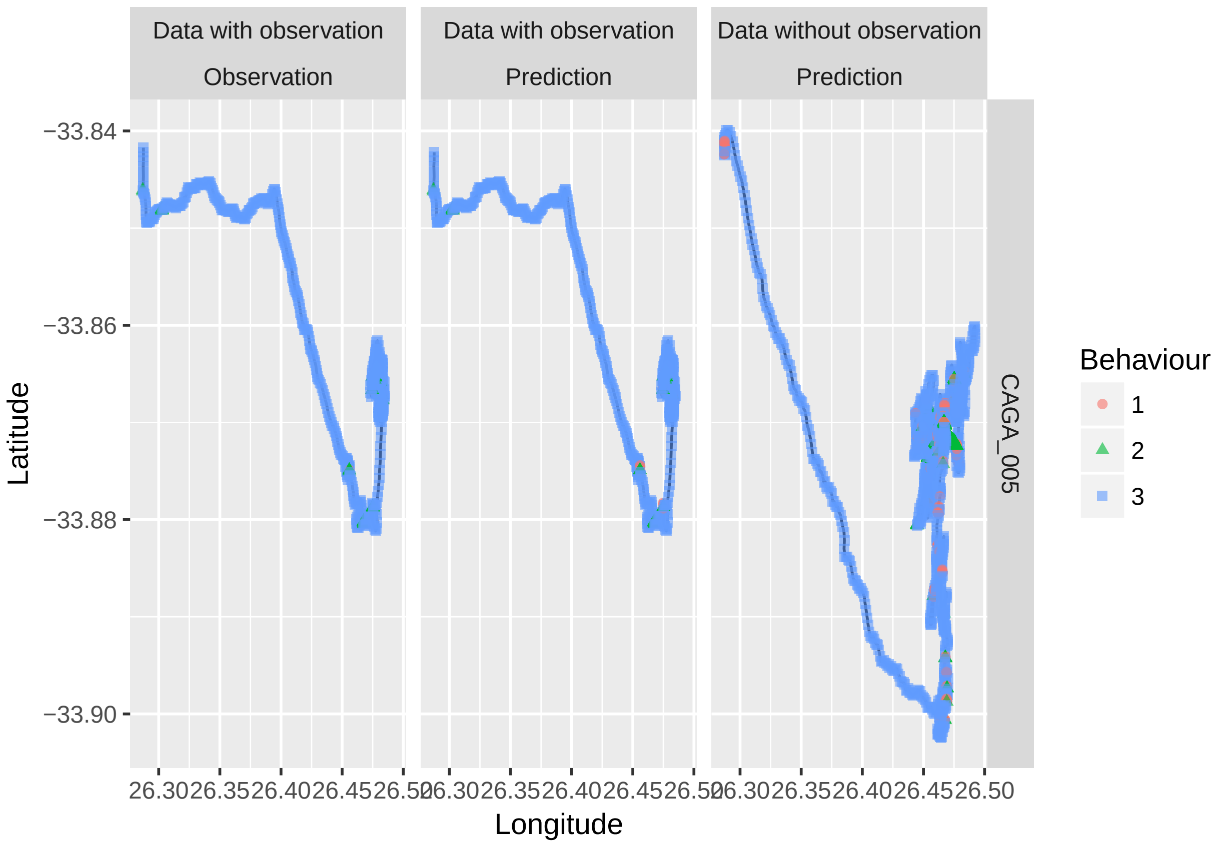
Results

The results regarding the behavioural states predicted vs the one observed are illustrated thanks to the `resB` functions.

```r
resB(xytb,"time",nob="-1")
```




```r
resB(xytb,"space",nob="-1")
```



Bibliography

Breiman, Leo. 2001. "Random Forests." *Machine Learning* 45: 5–32.

Calenge, Clément, Stéphane Dray, and Manuela Royer-Carenzi. 2008. "The Concept of Animals' Trajectories from a Data Analysis Perspective." *Ecological Informatics* In Press, Corrected Proof: –. http://www.sciencedirect.com/science/article/B7W63-4V28T4D-1/2/f122562e04e57b400dfd8b3858f30cf5.

Chen, Chao, Andy Liaw, and Leo Breiman. 2004. "Using Random Forest to Learn Imbalanced Data." Department of Statistics,UC Berkeley.