

# Package ‘m2b’

May 8, 2026

**Title** Movement to Behaviour Inference using Random Forest

**Version** 1.1.0

**Description** Prediction of behaviour from movement

characteristics using observation and random forest for the analyses of movement data in ecology.

From movement information (speed, bearing...) the model predicts the observed behaviour (movement, foraging...) using random forest. The model can then extrapolate behavioural information to movement data without direct observation of behaviours.

The specificity of this method relies on the derivation of multiple predictor variables from the movement data over a range of temporal windows. This procedure allows to capture as much information as possible on the changes and variations of movement and ensures the use of the random forest algorithm to its best capacity. The method is very generic, applicable to any set of data providing movement data together with observation of behaviour.

**VignetteBuilder** knitr

**Depends** R (>= 3.3.0)

**Imports** geosphere, caTools, ggplot2, randomForest, caret, methods, graphics, stats,

**Suggests** adehabitatLT, moveHMM, knitr, DiagrammeR, rmarkdown

**License** GPL-3

**URL** <https://github.com/ldbk/m2b>

**Encoding** UTF-8

**LazyData** true

**Date** 2025-06-24

**Maintainer** Laurent Dubroca <laurent.dubroca@gmail.com>

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Laurent Dubroca [aut, cre],  
Andréa Thiebault [aut]

**Repository** CRAN

**Date/Publication** 2025-06-25 12:20:20 UTC

## Contents

dxyt	2
dxyt2	3
extractRF	3
ltraj2xytb	4
m2b	5
modelRF	6
plot	7
resB	8
resRF	9
shiftvalue	10
track_CAGA_005	11
xytb	11
xytb-class	13
xytb2hmm	14
xytb2ltraj	15
<b>Index</b>	<b>17</b>

---

dxyt	<i>internal function</i>
------	--------------------------

---

### Description

internal function

### Usage

dxyt(xyt)

### Arguments

xyt           xyt parameters

### Author(s)

Laurent Dubroca

---

dxyt2	<i>internal function</i>
-------	--------------------------

---

**Description**

internal function

**Usage**

```
dxyt2(dxyt, winsize = seq(3, 13, 2), idquant = seq(0, 1, 0.25))
```

**Arguments**

dxyt	A parameter
winsize	A parameter
idquant	A parameter

**Author(s)**

Laurent Dubroca

---

extractRF	<i>Extract the random forest model from an xytb object</i>
-----------	--

---

**Description**

The random forest model is extracted from the xytb object to a randomForest object.

**Usage**

```
extractRF(xytb)
```

**Arguments**

xytb	an xytb object with a model.
------	------------------------------

**Value**

a randomForest object as defined by the randomForest package.

**Author(s)**

Laurent Dubroca

**See Also**

See [randomForest](#)

**Examples**

```
#track_CAGA_005 is dataset
#generate a complete xytb object with derived (over moving windows of 3, 5
#and 9 points, with quantile at 0, 50 and 100%) and shifted information on 10
#and 100 points
xytb<-xytb(track_CAGA_005,"a track",c(3,5,9),c(0,.5,1),c(10,100))
#compute a random forest model to predict behaviour (b, where -1 is
#unobserved behaviour) using the derived
#parameters ("actual")
xytb<-modelRF(xytb,"actual",nob="-1",colin=TRUE,varkeep=c("v","thetarel"),zerovar=TRUE)
#extract the model
modRF<-extractRF(xytb)
# results from randomForest package:
print(modRF)
plot(modRF)
```

---

ltraj2xytb

*ltraj object conversion to xytb object*


---

**Description**

The function converts a ltraj object to a xytb object (see <https://CRAN.R-project.org/package=adehabitatLT> and [xytb-class](#)).

**Usage**

```
ltraj2xytb(ltraj, desc = "ltraj object convert to xytb")
```

**Arguments**

ltraj	A ltraj object
desc	General descriptor of the data

**Value**

A xytb object with behavioural information taken from the infolocs (aka ltraj object should correspond the an export of xytb object in ltraj).

**Author(s)**

Laurent Dubroca

## Examples

```
## Not run:  
#track_CAGA_005 is a dataset  
xytb<-xytb(track_CAGA_005,"a track",3,.5)  
ltraj<-xytb2ltraj(xytb)  
xytb2<-ltraj2xytb(ltraj,"a track")  
  
## End(Not run)
```

---

m2b

*Movement to behaviour package*

---

## Description

A package to infer behaviour and social interaction from movement data

## Details

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 implementation (@breiman2001b, @chen2004). 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 package we developed a method to link the movement patterns of animals with their behaviour, using the random forest algorithm. The specificity of this method relies on the derivation of multiple predictor variables from the movement data over a range of temporal windows. This procedure allows to capture as much information as possible on the changes and variations of movement and ensures the use of the random forest algorithm to its best capacity. The method is very generic, applicable to any dataset providing movement data together with observation of behaviour.

## Author(s)

**Maintainer:** Laurent Dubroca <laurent.dubroca@gmail.com>

Authors:

- Andr ea Thiebault <andrea.thiebault@gmail.com>

**See Also**

Useful links:

- <https://github.com/lbdk/m2b>

---

 modelRF

*xytb randomForest function*


---

**Description**

Build a random forest model on a `xytb` object, predicting behaviour using only the variables calculated at the time of observation (type 'actual') or using the variable shifted backwards in time (type 'shifted'). Parameters are transferred to the `randomForest` or the `rfcv` functions of the `randomForest` package if needed.

**Usage**

```
modelRF(
  xytb,
  type = c("actual", "shifted"),
  nob = "-1",
  colin = TRUE,
  varkeep = c("v", "dist", "thetarel"),
  zerovar = TRUE,
  rfcv = FALSE,
  ntree = 501,
  importance = TRUE,
  ...
)
```

**Arguments**

<code>xytb</code>	an <code>xytb</code> object
<code>type</code>	character -actual or shifted- use actual data or shifted one to build the model
<code>nob</code>	character. Define the unobserved value of the behaviour (and where prediction are done)
<code>colin</code>	boolean - remove colinearity among predictors (see the <code>caret</code> package for more details)
<code>varkeep</code>	character vector - the variables names in this vector are kept in the model even if colinearity is found (usefull to keep 'classical' parameters and to help interpretation)
<code>zerovar</code>	boolean - remove near zero variance predictor (see the <code>caret</code> package for more details)

rfcv	boolean - run a random forest cross-validation for feature selection procedure for xytb (this call the rfcv function for the model). This function shows the cross-validated prediction performance of models with sequentially reduced number of predictors (ranked by variable importance) via a nested cross-validation procedure for a xytb object.
ntree	number of trees in the random Forest (see the randomForest package for more details)
importance	boolean (see the randomForest package for more details)
...	other arguements passed to randomForest or rfcv

**Author(s)**

Laurent Dubroca and Andréa Thiebault

**See Also**

See [randomForest](#) and [rfcv](#)

**Examples**

```
#track_CAGA_005 is dataset
#generate a complete xytb object with derived (over moving windows of 3, 5
#and 9 points, with quantile at 0, 50 and 100%) and shifted information on 10
#and 100 points
xytb<-xytb(track_CAGA_005,"a track",c(3,5,9),c(0,.5,1),c(10,100))
#compute a random forest model to predict behaviour (b, where -1 is
#unobserved behaviour) using the derived
#parameters ("actual")
xytb<-modelRF(xytb,"actual",nob="-1",colin=TRUE,varkeep=c("v","thetarel"),
zerovar=TRUE)
## Not run:
#cross-validation for the same model (time consuming !)
xytb<-modelRF(xytb,"actual",nob="-1",colin=TRUE,varkeep=c("v","thetarel"),
zerovar=TRUE,rfcv=TRUE)

## End(Not run)
```

---

plot

*xytb plot method*

---

**Description**

plot method for xytb object

**Usage**

```
## S4 method for signature 'xytb,missing'  
plot(x, y)
```

**Arguments**

x	An xytb object.
y	empty

**Value**

a plot

---

resB	<i>Representation of the predicted vs observed behaviour of an xytb object</i>
------	--

---

**Description**

Representation of the predicted vs observed behaviour of an xytb object

**Usage**

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

**Arguments**

xytb	an xytb object with predicted behaviour.
type	<ul style="list-style-type: none"><li>• time: plot results in time.</li><li>• space: plot results in space.</li><li>• density: plot results in space, adding density surface by behaviour.</li></ul>
nob	character. Define the unobserved value of the behaviour (and where prediction are done)

**Value**

a ggplot

**Author(s)**

Laurent Dubroca

**See Also**

See [randomForest](#)

## Examples

```
## Not run:
#track_CAGA_005 is dataset
#generate a complete xytb object with derived (over moving windows of 3, 5
#and 9 points, with quantile at 0, 50 and 100%) and shifted information on 10
#and 100 points
xytb<-xytb(track_CAGA_005,"a track",c(3,5,9),c(0,.5,1),c(10,100))
#compute a random forest model to predict behaviour (b, where -1 is
#unobserved behaviour) using the derived
#parameters ("actual")
xytb<-modelRF(xytb,"actual",nob="-1",colin=TRUE,varkeep=c("v","thetarel"),
zerovar=TRUE,rfcv=FALSE,step=.9)
#behaviour results:
resB(xytb,type="time",nob="-1")
resB(xytb,type="space",nob="-1")
resB(xytb,type="density",nob="-1")

## End(Not run)
```

---

resRF

*Random forest model outputs for a xytb object*


---

## Description

Diagnostic plots and tables for the random forest model used to predict behaviour on a xytb object (random forest convergence plot, variable importance plot, cross-validation plot, confusion matrix of the observed vs predicted behaviours).

## Usage

```
resRF(xytb, type = "rf")
```

## Arguments

xytb	An xytb object with a model.
type	<ul style="list-style-type: none"> <li>rf: plot of the OOB versus the number of trees (see <a href="#">randomForest</a>).</li> <li>importance: importance plot (see <a href="#">varImpPlot</a>).</li> <li>rfcv: plot of the cross-validated prediction performance of models with sequentially reduced number of predictors (see <a href="#">rfcv</a>).</li> <li>confusion: confusion matrix between observed and predicted behaviours (see <a href="#">confusionMatrix</a>).</li> </ul>

## Value

plots or tables.

**Author(s)**

Laurent Dubroca

**See Also**See [randomForest](#)**Examples**

```
## Not run:
#track_CAGA_005 is dataset
#generate a complete xytb object with derived (over moving windows of 3, 5
#and 9 points, with quantile at 0, 50 and 100%) and shifted information on 10
#and 100 points
xytb<-xytb(track_CAGA_005,"a track",c(3,5,9),c(0,.5,1),c(10,100))
#compute a random forest model to predict behaviour (b, where -1 is
#unobserved behaviour) using the derived
#parameters ("actual")
xytb<-modelRF(xytb,"actual",nob="-1",colin=TRUE,varkeep=c("v","thetarel"),
zerovar=TRUE,rfcv=TRUE,step=.9)
#modelling results
resRF(xytb,type="rf")
resRF(xytb,type="importance")
resRF(xytb,type="rfcv")
resRF(xytb,type="confusion")

## End(Not run)
```

---

 shiftvalue

*internal function*


---

**Description**

internal function

**Usage**

```
shiftvalue(dat0, mov = seq(5, 250, 5))
```

**Arguments**

dat0	A parameter
mov	A parameter

**Author(s)**

Laurent Dubroca

---

track_CAGA_005	<i>Data collected from a cape gannet track (Morus capensis, Lichtenstein 1823), breeding on Bird Island (Algo Bay, South Africa) in december 2010.</i>
----------------	--

---

### Description

Tracking data recorded with a GPS device (i-GotU GT-600, Mobile Action Technology Inc., Taipei, Taiwan). Regular step duration at 5s. Behaviour data recorded with a video camera (Camsports nano, CamsportsTM, Estrablin, France). Observations of taking off, landing and diving, from which three behavioural states were inferred: flying, sitting on the water, diving. Recording during the first 90 min (short autonomy of camera) of the trip.

### Usage

track\_CAGA\_005

### Format

Tracking data

Columns:

x = longitude in decimal degrees

y = latitude in decimal degrees

t = time in POSIXct

b = behaviour observed on video data (3:flying, 2:sitting on water, 1:diving ,-1:no data)

id = individual id

### Source

Andréa Thiebault

---

xytb	<i>xytb class constructor</i>
------	-------------------------------

---

### Description

The methods to build an xytb object (see [xytb-class](#) for the class description).

**Usage**

```

xytb(object, desc, winsize, idquant, move, ...)

## S4 method for signature 'missing,missing,ANY,ANY,ANY'
xytb(desc = "null xybt object")

## S4 method for signature 'data.frame,character,ANY,ANY,ANY'
xytb(object, desc, winsize, idquant, move, ...)

## S4 method for signature 'data.frame,character,vector,vector,ANY'
xytb(object, desc, winsize, idquant, move, ...)

## S4 method for signature 'data.frame,character,vector,vector,vector'
xytb(
  object,
  desc = "unknow track",
  winsize = seq(3, 13, 2),
  idquant = seq(0, 1, 0.25),
  move = c(5, 10),
  ...
)

```

**Arguments**

object	a data frame with 5 columns															
	<table> <tr> <td>id</td> <td>individual id</td> <td>character</td> </tr> <tr> <td>x</td> <td>longitude</td> <td>decimal degree</td> </tr> <tr> <td>y</td> <td>latitude</td> <td>decimal degree</td> </tr> <tr> <td>t</td> <td>date and time</td> <td>POSIXct</td> </tr> <tr> <td>b</td> <td>behaviour</td> <td>character</td> </tr> </table>	id	individual id	character	x	longitude	decimal degree	y	latitude	decimal degree	t	date and time	POSIXct	b	behaviour	character
id	individual id	character														
x	longitude	decimal degree														
y	latitude	decimal degree														
t	date and time	POSIXct														
b	behaviour	character														
desc	vector of character describing the dataset															
winsize	a numerical vector giving the length of the windows used to calculate moving standard deviation, average, mad and quantile for the speed (v), the distance (dist) and the relative angle (thetarel).															
idquant	a numerical vector giving the quantiles to be calculated. For example if idquant=c(0,0.25,.5,1), the quantiles at 0% (min), 25% (first quartile), 50% (median) and 100% (max) will be calculated.															
move	a numerical vector providing the shift used to computed parameters back in time. For example if move=c(5,10,100), the parameters will be shifted backward by 5, 10 and 100 locations. Optional.															
...	part of the generic definition															

**Value**

an xytb object

**Methods' signature**

- `xytb()`: generate an empty `xytb` object.
- `xytb(data.frame, character vector)`: generate an `xytb` object with track information (only slot `desc`, `xyt` and `b` are computed).
- `xytb(data.frame, character vector, numerical vector, numerical vector, numerical vector)`: generate an `xytb` object with track information (slots `desc`, `xyt`, `b`) and derived information `dxyt` and `befdxyt`. `dxyt` contains statistical derivatives of speed, distance and relative angle calculated on moving windows given by the `winsize` parameters. Statistical derivatives are standard deviation, mean, median absolute deviation and quantiles. Quantiles are defined by the `idquant` parameters. `befdxyt` contains `dxyt` values shifted back in time according to the `move` parameters.

**Author(s)**

Laurent Dubroca

**Examples**

```
#generate an empty xytb object
xytb()
#generate an xytb object with track information only
#track_CAGA_005 is a dataset
simplexytb<-xytb(track_CAGA_005,"a track")
#generate a complete xytb object with derived (over moving windows of 3, 5
#and 9 points, with quantile at 0, 50 and 100%) and shifted information on 10
#and 100 points
xytb<-xytb(track_CAGA_005,"a track",c(3,5,9),c(0,.5,1),c(10,100))
```

---

xytb-class

*xytb class definition*

---

**Description**

`xytb` is a trajectory object with observed behaviour

**Slots**

`desc` a character vector. A description of the data.

`xyt` a data frame of the track information. One row is one position described by 11 variables :

<code>id:</code>	individual id
<code>t:</code>	time in POSIXct
<code>x:</code>	longitude
<code>y:</code>	latitude
<code>dt:</code>	time (s) between the current location and the next one

dist: distance (m) between the current location and the next one calculated using the Vincenty ellipsoid method. See [dist](#)  
 v: speed (m.s<sup>-1</sup>)  
 dx: the increase of the move in the x direction.  
 dy: the increase of the move in the y direction.  
 theta: the angle between each move and the x axis.  
 thetare1: the turning angle between successive moves.

This slot is very similar to the `ltraj` class of the `adehabitatLT` package.

`b` a data frame of the observed behaviour related to the track. One row is one behavioural observation related to the track described by 3 variables:

```

id: individual id
t:  time in POSIXct
b:  observed behaviour (character)
  
```

`dxyt` a data frame of the derived variables from the track.

`befdxyt` a data frame of the derived variables shifted back in time.

`model` a `randomForest` model (see [randomForest](#)).

`rfcv` the output of the cross-validation procedure (see [rfcv](#)).

`predb` a data frame of the predicted behaviour using the random forest model of the slot `model`. 3 variables :

```

id: individual id
t:  time in POSIXct
b:  observed behaviour (character)
  
```

### Author(s)

Laurent Dubroca

---

xytb2hmm

*xytb object conversion to moveHMM object*

---

### Description

The function converts a `xytb` object to a `moveHMM` dataframe (see <https://CRAN.R-project.org/package=moveHMM> and [prepData](#)).

### Usage

```
xytb2hmm(xytb)
```

**Arguments**

xytb            A xytb object

**Value**

A dataframe ready to be used by the function of the ‘moveHMM’

**Author(s)**

Laurent Dubroca

**Examples**

```
#track_CAGA_005 is a dataset
xytb<-xytb(track_CAGA_005,"a track",3,.5)
xyhmm <-xytb2hmm(xytb)
#then you can use moveHMM functions
if(requireNamespace("moveHMM")){
  plot(xyhmm)
}
```

---

xytb2ltraj

*xytb class conversion to ltraj object*

---

**Description**

The function converts a xytb object to a ltraj object as defined in the adehabitatLT package (see <https://CRAN.R-project.org/package=adehabitatLT> as.ltraj).

**Usage**

```
xytb2ltraj(xytb)
```

**Arguments**

xytb            An xytb object

**Value**

A ltraj object with behavioural information recorded in the infoloc

**Author(s)**

Laurent Dubroca

**Examples**

```
#track_CAGA_005 is a dataset
xytb<-xytb(track_CAGA_005,"a track",3,.5)
ltraj<-xytb2ltraj(xytb)
#all adehabitatLT function are now available
summary(ltraj)
plot(ltraj)
```

# Index

- \* **datasets**
  - track\_CAGA\_005, [11](#)
- as.ltraj, [15](#)
- confusionMatrix, [9](#)
- distVincentyEllipsoid, [14](#)
- dxyt, [2](#)
- dxyt2, [3](#)
  
- extractRF, [3](#)
  
- ltraj2xytb, [4](#)
  
- m2b, [5](#)
- m2b-package (m2b), [5](#)
- modelRF, [6](#)
  
- plot, [7](#)
- plot,xytb,missing-method (plot), [7](#)
- prepData, [14](#)
  
- randomForest, [4](#), [7–10](#), [14](#)
- resB, [8](#)
- resRF, [9](#)
- rfcv, [7](#), [9](#), [14](#)
  
- shiftvalue, [10](#)
  
- track\_CAGA\_005, [11](#)
  
- varImpPlot, [9](#)
  
- xytb, [11](#)
- xytb,data.frame,character (xytb), [11](#)
- xytb,data.frame,character,ANY,ANY,ANY-method (xytb), [11](#)
- xytb,data.frame,character,vector,vector (xytb), [11](#)
- xytb,data.frame,character,vector,vector,ANY-method (xytb), [11](#)
- xytb,data.frame,character,vector,vector,vector (xytb), [11](#)
- xytb,data.frame,character,vector,vector,vector-method (xytb), [11](#)
- xytb,missing,missing,ANY,ANY,ANY-method (xytb), [11](#)
- xytb-class, [4](#), [11](#), [13](#)
- xytb2hmm, [14](#)
- xytb2ltraj, [15](#)