`foieGras` an R package for animal movement data: rapid quality control, behavioural estimation and simulation

Ian D. Jonsen[1,*] and Toby A. Patterson[2]

[1]Department of Biological Sciences, Macquarie University, Sydney, NSW, Australia
[2]CSIRO Ocean and Atmosphere Research, Hobart, TAS, Australia
[*]corresponding author, ian.jonsen@mq.edu.au

Abstract

text…

Keywords:

1 | Introduction

The R package `foieGras`, pronounced "*fwah grah*", …

2 | `foieGras` Overview

The workflow for `foieGras` is deliberately simple, with much of the usual track data processing checks and formatting handled automatically. The main functions are listed in Table 1. When fitting a model, `foieGras` automatically detects the type of tracking data location quality classes designations that are typical of Argos data and that can be added to the data by the researcher for other types of track data. Based on the location quality classes and other, optional information on observation errors contained in the data, `foieGras` chooses an appropriate measurement error model for each observation. This capability allows for combinations of different tracking data types, e.g., Argos and GPS, in a single input data frame and to be fit in a single state-space model.

2.1 | Data preparation

Animal tracking data, consisting of a time-series of location coordinates, can be read into R as a data frame using standard functions such as `read.csv`. The canonical data format for Argos tracks consists of a data frame with 5 columns corresponding to the following named variables: `id` (individual id), `date` (date and time), `lc` (location class), `lon` (longitude), `lat` (latitude). Optionally, an additional 3 columns, `smaj` (semi-major axis), `smin` (semi-minor axis), `eor` (ellipse orientation), providing Argos error ellipse information may be included.

Other types of track data can be accommodated, for example, by including the `lc` column where all `lc = "G"` for GPS data. In this case, measurement error in the GPS locations is assumed to have a standard deviation of 0.1 x Argos class 3 locations (approximately 30 m). Other types of track data can be considered in a similar manner (see the package vignette for further details).

Table 1: Main functions for the R package `foieGras`

Function	Description
`fit_mpm`	Fit a Move Persistence Model to location data
`fit_ssm`	Fit a State-Space Model to location data
`fmap`	Plot fitted/predicted locations on a map with or without a defined projection
`grab`	Extract fitted/predicted/observed locations from a foieGras model, with or without projection information
`osar`	Estimate One-Step-Ahead Residuals from a foieGras SSM
`plot.fG_mpm`	Plot move persistence estimates as 1-D or 2-D (along track) time-series
`plot.fG_osar`	Plot One-Step-Ahead Residuals from a foieGras SSM
`plot.fG_ssm`	Visualise the fit of a foieGras SSM to data

2.2 | State-space model fitting - `fit_ssm`

State-space models are fit using `fit_ssm`. There are a large number of options that can be set in `fit_ssm` (see Suppl for details). We focus only the essential options here:

- `data` the input data structured as described in **2.1**
- `vmax` a maximum threshold speed (ms^{-1}) to help identify potential outlier locations
- `model` the process model to be used
- `time.step` the prediction time interval (h)

The function first invokes an automated data processing stage where the following occurs: 1) data type (Argos Least-Squares, Argos Kalman Filter/Smoother, GPS, or General (e.g., processed light-level geolocations, acoustic telemetry, coded VHF telemetry) is determined; 2) datetimes are converted to POSIXt format, chronological order is ensured, and duplicate datetime records are removed; 3) observations occurring less than `min.dt` seconds after a prior observation are removed; 4) a speed filter (`sda` from the `trip` R package; Sumner et al., 2009) is used to identify potential outlier locations; 5) locations are projected from spherical lon-lat coordinates to planar x,y coordinates in km.

The function then fits a state-space model to the processed data, where the process model (currently, either a continuous-time `rw` or a continuous-time `crw`) is specified by the user and the measurement model(s) are selected automatically (see Jonsen et al., 2020 for model details). The model is fit by numerical optimization of the likelihood using either the `optim` or `nlminb` R function. The R package `TMB`, Template Model Builder (Kristensen et al., 2016), is used to compute the gradient function in C++ via reverse-mode auto-differentiation and the Laplace Approximation is used to integrate out the latent states (random effects). Fits to a single versus multiple individuals are handled automatically, with sequential SSM fits occurring in the latter case. No hierarchical or pooled estimation among individuals is currently available.

`fit_ssm` returns a `foieGras` fit object (a nested data frame with class `fG_ssm`). The outer data frame lists the individual id(s), basic convergence information and a list with class `ssm`. This list contains dense information on the model parameter and state estimates, predictions, processed data, optimizer results, and other diagnostic and contextual information. Users can extract a simple data frame of SSM fitted (location estimates corresponding to the, typically irregular, observation times) or predicted values (locations predicted at regular `time.step` intervals) using the `grab` function.

2.3 | Model checking and visualisation - `osar`, `plot`, `fmap`

Before using fitted or predicted locations, a model fit should be checked and visualised to confirm that the model adequately describes the data. In linear regression and a variety of analogous methods, goodness-of-fit can be assessed by calculating standard residuals such as Pearson or deviance residuals. There is no simple way to calculate residuals for latent variable models that have non-finite state-spaces and that may be nonlinear, but they can be computed based on iterative forecasts of the model (Thygesen et al., 2017). The `osar` function computes one-step-ahead (prediction) residuals and uses the `oneStepPredict` function from the `TMB` R package to make this as efficient as possible. A set of residuals are calculated for the `x` and `y` values corresponding to the fitted values from the SSM and returned as an `fG_osar` object.

A generic `plot` method provides an easy way to visualise the `fG_osar` residuals. Time-series plots of the prediction residuals can be used to detect temporal changes in goodness-of-fit. Quantile-quantile plots of residuals against standard normal quantiles can be used to detect departures from normality. Sample autocorrelation function plots of the residuals are useful for detecting auto-correlation not accounted for by the model. Assessing residual autocorrelation can be particularly important as Argos locations, for example, are themselves derived from a time-series model (Lopez et al., 2015) which can introduce additional autocorrelation in the location errors.

State-space model fits to data can also be visualised by using the generic `plot` function on an `fG_ssm` data frame. Options exist to plot fitted or predicted values along with observations as either paired, 1-D time-series or as 2-D tracks with confidence intervals or ellipses, respectively. These plots provide a more intuitive and rapid method for assessing SSM fits to data, however, they do not replace the residual diagnostics. Fitted `fG_ssm` data frames can be mapped using the `fmap` function for single or multiple individuals. Estimated tracks can be displayed with or without confidence ellipses, observations, and/or a projection and maps of single tracks can be coloured by date.

2.4 | Behavioural estimation - `fit_mpm`

The `fit_mpm` function fits a simple move persistence model to estimate a continuous-valued, time-varying latent variable that indexes changes in movement behaviour (Jonsen et al., 2019). This variable measures the autocorrelation in speed and direction between consecutive pairs of movements such that high values correspond to fast, directed movements at one end of the continuum and low values correspond to slow, tortuous movements at the other end. It's important to note that this approach is unlike hidden Markov models (McClintock & Michelot, 2018; Michelot et al., 2016) and some state-space models (Jonsen, 2016) as there is no notion of discrete behavioural states that animals periodically switch between. Nonetheless, move persistence can be used to identify objectively places where animals spend disproportionately more or less time, and with extensions be correlated with environment or other covariates (See Examples 3.x).

`fit_mpm` assumes that locations are absent of measurement error and takes either a `fG_ssm` data frame as input or a data frame with the follow variables: `id`, `date`, `x`, `y`, where `x` and `y` coordinates can be planar `x,y` or spherical `long,lat`. This latter input format allows the model to be fit easily to GPS or other tracking data with negligible measurement error. When the data contain multiple individuals, the default model is fit jointly by assuming all individuals share the same move persistence variance parameter. There is an option to fit the model separately to each individual. The time-series of estimated move persistence with confidence intervals can be visualised by using the generic `plot` function with the resulting `fG_mpm` data frame. Visualisation of move persistence along the 2-D tracks can be plotted or mapped by using the `plot` or `fmap` functions, respectively, and supplying both the `fG_mpm` and `fG_ssm` data frames.

2.5 | Simulation - `simulate`

3 | Examples

Southern Elephant seal

something quick about the data here. Sentence about fitting `rw` and `crw` models then show code for fitting:

```
fit.rw <- fit_ssm(ellie, vmax=4, model="rw", time.step=12, verbose=0)
```

```
res.rw <- osar(fit.rw)

fit.crw <- fit_ssm(ellie, vmax=4, model="crw", time.step=12, verbose=0)
```

```
res.crw <- osar(fit.crw)
```

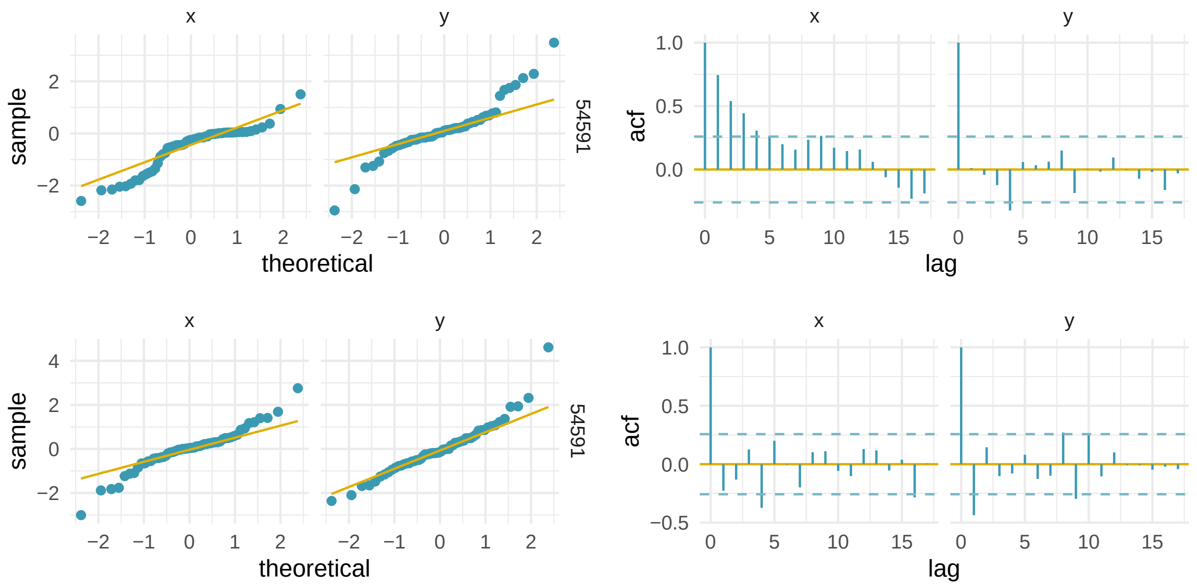

Figure 1: Ellie ex.

3.x | Extending the behavioural model using `mpmm`

4 | Discussion

Acknowledgements

We thank xx, xx for providing data used in the examples, and xx, xx for helpful comments on an earlier draft of this manuscript. We thank Marie Auger-Méthé for contributing code to the movement persistence models. IDJ acknowledges support from a Macquarie University co-Funded Fellowship and from partners: the US Office of Naval Research, Marine Mammal Program (grant N00014-18-1-2405); the Integrated Marine Observing System (IMOS); Taronga Conservation Society; the Ocean Tracking Network; Birds Canada; and Innovasea/VEMCO. TAP was supported by CSIRO Oceans & Atmosphere internal research funding scheme.

Author's Contributions

IDJ developed the R package; IDJ and TAP developed the state-space models and wrote the manuscript.

Data Accessibility

All code mentioned here is provided in the `foieGras` package for R available on CRAN at https://CRAN.R-project.org/package=foieGras. The development version of the package is available on GitHub at https://github.com/ianjonsen/foieGras. Data used in the examples are available at…

ORCID

Ian D Jonsen https://orcid.org/0000-0001-5423-6076
Toby A Patterson https://orcid.org/0000-0002-7150-9205

References

Jonsen, I. D. (2016). Joint estimation over multiple individuals improves behavioural state inference from animal movement data. *Scientific Reports*, *6*, 20625.

Jonsen, I. D., Patterson, T. A., Costa, D. P., Doherty, P. D., Godley, B. J., Grecian, W. J., Guinet, C., Hoenner, X., Kienle, S. S., Robinson, P. W., Votier, S. C., Whiting, S., Witt, M. J., Hindell, M. A., Harcourt, R. G., & McMahon, C. R. (2020). A continuous-time state-space model for rapid quality control of Argos locations from animal-borne tags. *Movement Ecology*, *8*, 31.

Jonsen, I., McMahon, C. R., Patterson, T. A., Auger-Méthé, M., Harcourt, R., Hindell, M. A., & Bestley, S. (2019). Movement responses to environment: Fast inference of variation among southern elephant seals with a mixed effects model. *Ecology*, *100*, e02566.

Kristensen, K., Nielsen, A., Berg, C. W., Skaug, H., & Bell, B. M. (2016). TMB: Automatic differentiation and Laplace approximation. *Journal of Statistical Software*, *70*, 1–21.

Lopez, R., Malardé, J., Danès, P., & Gaspar, P. (2015). Improving Argos Doppler location using multiple-model smoothing. *Animal Biotelemetry*, *3*, 32.

McClintock, B. T., & Michelot, T. (2018). MomentuHMM: R package for generalized hidden Markov models of animal movement. *Methods in Ecology and Evolution*, *9*, 1518–1530.

Michelot, T., Langrock, R., & Patterson, T. A. (2016). MoveHMM: An R package for the statistical modelling of animal movement data using hidden Markov models. *Methods in Ecology and Evolution*, *7*, 1308–1315.

Sumner, M. D., Wotherspoon, S. J., & Hindell, M. A. (2009). Bayesian estimation of animal movement from archival and satellite tags. *PLoS ONE*, *4*(10). https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0007324

Thygesen, U. H., Albertsen, C. M., Berg, C. W., Kristensen, K., & Neilsen, A. (2017). Validation of ecological state space models using the Laplace approximation. *Environmental and Ecological Statistics*. https://doi.org/DOI:%2010.1007/s10651-017-0372-4