

# aniMotum, an R package for animal movement data: Rapid quality control, behavioural estimation and simulation

Ian D. Jonsen<sup>1</sup>  | W. James Grecian<sup>2</sup>  | Lachlan Phillips<sup>1</sup>  | Gemma Carroll<sup>3</sup>  |  
Clive McMahon<sup>4</sup>  | Robert G. Harcourt<sup>1</sup>  | Mark A. Hindell<sup>5</sup>  | Toby A. Patterson<sup>6</sup> 

<sup>1</sup>School of Natural Sciences, Macquarie University, Sydney, New South Wales, Australia; <sup>2</sup>Sea Mammal Research Unit, Scottish Oceans Institute, University of St Andrews, St Andrews, UK; <sup>3</sup>Environmental Defense Fund, Seattle, Washington, USA; <sup>4</sup>Sydney Institute of Marine Science, Mosman, New South Wales, Australia; <sup>5</sup>Institute for Marine and Antarctic Studies, University of Tasmania, Hobart, Tasmania, Australia; <sup>6</sup>CSIRO Oceans and Atmosphere, Hobart, Tasmania, Australia

## Correspondence

Ian D. Jonsen  
Email: [ian.jonsen@mq.edu.au](mailto:ian.jonsen@mq.edu.au)

## Present address

W. James Grecian, Department of Geography, Durham University, Durham, Durham, UK

## Funding information

Office of Naval Research, Grant/Award Number: N00014-18-1-2405; CSIRO; CNES-TOSCA; Australia's Integrated Marine Observing System (IMOS); Australian Research Council, Grant/Award Number: LP160100162; Fisheries and Oceans Canada; NERC Arctic Office, Grant/Award Number: NE/P00623X/1 and NE/P006035/1; UKRI Natural Environment Research Council; German Federal Ministry of Education and Research

**Handling Editor:** Edward Codling

## Abstract

1. Animal tracking data are indispensable for understanding the ecology, behaviour and physiology of mobile or cryptic species. Meaningful signals in these data can be obscured by noise due to imperfect measurement technologies, requiring rigorous quality control as part of any comprehensive analysis.
2. State-space models are powerful tools that separate signal from noise. These tools are ideal for quality control of error-prone location data and for inferring where animals are and what they are doing when they record or transmit other information. However, these statistical models can be challenging and time-consuming to fit to diverse animal tracking data sets.
3. The R package aniMotum eases the tasks of conducting quality control on and inference of changes in movement from animal tracking data. This is achieved via: (1) a simple but extensible workflow that accommodates both novice and experienced users; (2) automated processes that alleviate complexity from data processing and model specification/fitting steps; (3) simple movement models coupled with a powerful numerical optimization approach for rapid and reliable model fitting.
4. We highlight aniMotum's capabilities through three applications to real animal tracking data. Full R code for these and additional applications is included as Supporting Information, so users can gain a deeper understanding of how to use aniMotum for their own analyses.

## KEYWORDS

animal movement, biologging, bio-telemetry, move persistence, movement behaviour, random walk, simulation, state-space model

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial](https://creativecommons.org/licenses/by-nc/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2023 The Authors. *Methods in Ecology and Evolution* published by John Wiley & Sons Ltd on behalf of British Ecological Society.

## 1 | INTRODUCTION

The collection of animal telemetry and bio-logging data has become essential for understanding the movements, behaviour, social interactions, foraging ecology, physiology, habitat use and population dynamics of mobile and/or cryptic species (Kays et al., 2015). In addition, the sophistication and precision of current technology now enable the use of tagged animals to collect in situ environmental information complementing data collected from traditional Earth observing platforms (McMahon et al., 2021). However, in each of these applications, common issues with tracking data such as irregularly timed measurements, sensor biases and location measurement errors must be overcome.

State-space models (SSMs) are powerful tools for conducting quality control of and making behavioural inference from animal tracking data (Patterson et al., 2008). Within this framework, an animal's true location and/or behaviour in the wild are unobserved states, while measurements recorded by animal-borne sensors provide noisy observations. The SSM comprises both a movement process model (e.g. a correlated random walk) and a measurement model that are fit to animal tracks to estimate true location and behaviour states, thereby separating ecologically meaningful signals from the noise of sensor measurements.

Here, we introduce aniMotum (a shorthand of the Latin for *animal movement*), a package for fitting SSMs in R (R Core Team, 2021). This package has two aims: (1) to be a simple and fast implementation

of SSMs to quality control error-prone animal location data (Jonsen et al., 2020); and (2) for inference of changes in movement behaviour along animal tracks (Jonsen et al., 2019). Although several implementations of SSMs for animal movement data exist, for example, bsam (Jonsen et al., 2005), crawl (Johnson et al., 2008) and ctm (Calabrese et al., 2016), their broad utility can be hampered by their user interfaces that tend to be technically complex. aniMotum has a more user-friendly implementation suitable for novices, while experienced users can exert control over many aspects of the package functions via optional arguments.

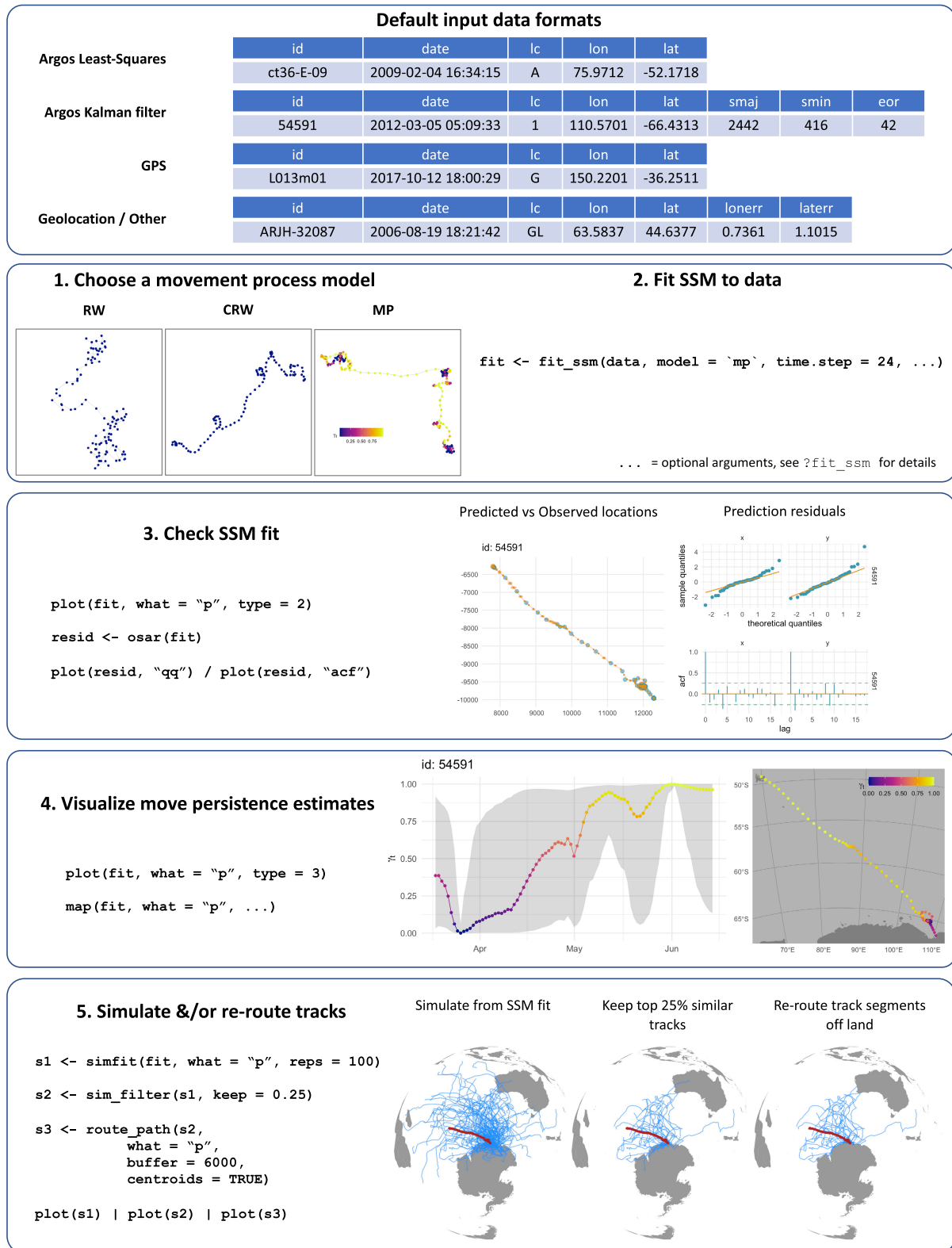
We describe the main features of aniMotum and illustrate its use through a set of applications drawing on Argos and GPS tracking data. Full R code and data for each of the applications are provided in the Supporting Information. Additional details on package functions and usage can be found in the help files and vignettes (<https://ianjonsen.github.io/aniMotum/index.html>).

## 2 | ANIMOTUM OVERVIEW

The workflow for aniMotum is deliberately simple, with many of the usual track data processing checks handled automatically. Here, we outline the main aspects of the aniMotum package. The package's main functions with brief code examples are listed in Table 1, and a generalized workflow with example code is depicted in Figure 1.

**TABLE 1** Main aniMotum functions with description and example code. In the code examples, ... denotes optional arguments. Details on these arguments can be found in the package help files for each function (e.g. ?fit\_ssm())

Function	Description	Code
format_data()	Coerce tracking data into the format expected by 'fit_ssm()	data <- format_data(x, ...)
fit_ssm()	Fit a state-space model to location data	fit <- fit_ssm(data, model, time.step, ...)
fit_mpm()	Fit a move persistence model to location data	fmp <- fit_mpm(fit, model, ...)
grab()	Extract fitted/predicted/observed locations from a aniMotum model, with or without projection information	locs <- grab(fit, what = 'predicted', ...)
osar()	Estimate One-Step-Ahead Residuals from a aniMotum SSM	res <- osar(fit, ...)
map()	Map fitted/predicted locations with or without a defined projection	map(fit, fmp, what = 'predicted', ...)
sim()	Simulate individual animal tracks with Argos LS or KF errors	slocs <- sim(N, model, ...)
sim_fit()	Simulate animal tracks from SSM model fit objects	sfit <- simfit(fit, ...)
sim_filter()	Filter simulated tracks according to similarity criteria	sfit <- sim_filter(sfit, keep, ...)
route_path()	Reroute path so estimated locations are off land	fit <- route_path(fit, what = 'predicted', ...)
plot.ssm_df()	Plot the fit of a aniMotum SSM to data	plot(fit, what = 'predicted', ...)
plot.osar()	Plot One-Step-Ahead Residuals from a aniMotum SSM	plot(res, type = 'ts')



**FIGURE 1** A generalized aniMotum workflow showing default input data formats, model fitting, model checking/validation, visualization of movement behaviour estimates, track simulation and re-routing around land. Example code is displayed for each stage. Details on default input data formats and options for handling other formats are provided in the Overview vignette.

## 2.1 | Data input—format\_data()

Animal tracking data, consisting of a time series of location coordinates, can be read into R as a data.frame or tibble using standard functions such as read.csv or read\_csv. Possible default input data formats are shown in Figure 1. The format\_data() function can be used to coerce non-default data formats into that expected by fit\_ssm(). format\_data() can be used explicitly in a workflow or it will be called automatically by fit\_ssm(). More details on input data formats and the use of format\_data() are provided in the Overview vignette (<https://ianjensen.github.io/aniMotum/articles/Overview.html>). Additionally, aniMotum can accept data as an sf-tibble or sf data.frame with locations in any recognized projection, in which case the SSM-estimated locations will be returned in the same projection.

## 2.2 | State-space model fitting—fit\_ssm()

State-space models are fit using fit\_ssm() (Figure 1). Data type is detected from the location quality classes typical of Argos data; classes can be added for other data types. Based on the location quality classes and optional information on measurement errors contained in the data, an appropriate measurement error model is selected for each observation (Jonsen et al., 2020). Fits to single versus multiple individuals are handled automatically, with sequential fitting for the latter. No hierarchical or pooled estimation among individuals is

currently possible. aniMotum' process and measurement error models are outlined in Table 2.

Many optional arguments can be specified in fit\_ssm() and are explained in the package documentation. We focus on the essential ones here:

- data—the input data as described in Section 2.1 and Figure 1
- model—the process model to be used (one of rw, crw or mp; see Table 1 for details)
- time.step—the prediction time interval (expressed in decimal hours)

The function invokes an automated data processing stage where the data type and measurement error model(s) are determined, observation times are sorted and checked for duplicates and a speed filter identifies extreme locations to be ignored by the SSM. The SSM is then fitted to the processed data, with the user-specified process model and automatically selected measurement error model(s). The R package TMB, Template Model Builder (Kristensen et al., 2016), is used to compute the gradient function in C++, which greatly speeds model fitting. A fit object is returned as a nested tibble, listing the individual animal id(s), basic convergence information and a list of model output including estimated parameters and states, processed data and diagnostic information. A data.frame of SSM fitted (location estimates corresponding to the observation times) or predicted values (locations predicted at regular time.step intervals) can be extracted using the grab() function. Parameter estimates, AIC and

**TABLE 2** aniMotum movement process and measurement error models with brief descriptions and parameters. The process and measurement error models are the two components of a state-space model. Parameters listed are those estimated when fitting the state-space model to data, these estimates can be accessed via summary() (see Overview vignette for details)

Model	Description	Parameters
Process models		
rw—Random walk	Movements are random in direction and magnitude.	$\rho_p, \sigma_x, \sigma_y$
crw—Correlated random walk	Movements are random and correlated in direction and magnitude.	D
mp—Move persistence	Movements are random with correlation in direction and magnitude that varies in time.	$\rho_p, \sigma_x, \sigma_y, \sigma_g$
Measurement error models		
LS—Argos least-squares	Location measurement errors assigned via Argos location quality class and error multiplication factors (see ?emf for details).	$\rho_o, \tau_x, \tau_y$
KF—Argos Kalman filter	Location measurement errors specified by error ellipse variables (smaj, smin, eor) present in data (see Overview vignette for details).	$\psi$
G—GPS	Location measurement errors are assumed by default to be 10% of Argos location quality class 3 (see ?emf for details).	$\rho_o, \tau_x, \tau_y$
GL—Geolocation	Location measurement errors are specified by lonerr, laterr variables present in data.	$\rho_o, \tau_x, \tau_y$

Note:  $\rho_p$ —correlation between x and y process variances (rho\_p).  $\sigma_x, \sigma_y$ —process standard deviation in x and y directions respectively (sigma\_x, sigma\_y). D—process variance in x and y directions (D).  $\sigma_g$ —standard deviation of random walk on m  $\gamma_t$  (sigma\_g).  $\rho_o$ —correlation between x and y measurement error variances (rho\_o).  $\tau_x, \tau_y$ —standard deviation of measurement errors in x and y directions respectively (tau\_x, tau\_y).  $\psi$ —scaling parameter to account for potential bias in semi-minor axis of Argos KF error ellipses (psi).

other information are viewed using the `summary()` function. Further details on SSM fitting are here: ([https://ianjensen.github.io/aniMotum/articles/SSM\\_fitting.html](https://ianjensen.github.io/aniMotum/articles/SSM_fitting.html)).

### 2.3 | Behavioural estimation—`fit_ssm()`, `fit_mpm()`

Move persistence, an index of movement behaviour, can be estimated as a continuous-valued (0–1), time-varying latent variable that represents changes in movement pattern based on autocorrelation in speed and direction (Jonsen et al., 2019). There are two approaches in aniMotum for estimating move persistence. The first is to use `fit_ssm()` with `model = mp`, which fits a continuous-time move persistence model in state–space form and thereby simultaneously estimates true locations and move persistence from the error-prone telemetry data (Figure 1). This approach is most appropriate for fitting to irregularly timed and error-prone Argos data as both aspects are taken into account explicitly. The second is to use `fit_mpm()`, which can take as input either location data or SSM-estimated locations from an `fit_ssm()` model fit object. This approach is generally more appropriate when the data have minimal measurement error (e.g. GPS locations; see Application 3.2). Further details on fitting move persistence models are here: [https://ianjensen.github.io/aniMotum/articles/Move\\_persistence\\_models.html](https://ianjensen.github.io/aniMotum/articles/Move_persistence_models.html).

### 2.4 | Model checking and visualization—`osar()`, `plot()`, `map()`

Before using fitted or predicted locations, a `fit_ssm()` model fit should be checked and visualized to confirm that the model adequately describes the data. The `osar()` function computes one-step-ahead (prediction) residuals via the `oneStepPredict()` function from the TMB R package (Kristensen et al., 2016; Thygesen et al., 2017). A set of residuals are calculated for the *x* and *y* values corresponding to the fitted values from the SSM. A generic `plot()` (`plot.osar`) method provides an easy way to visualize the prediction residuals as time-series plots, quantile–quantile plots or autocorrelation functions (Figure 1). Further details on model validation are here: [https://ianjensen.github.io/aniMotum/articles/SSM\\_validation.html](https://ianjensen.github.io/aniMotum/articles/SSM_validation.html).

State–space model fits to data can also be visualized by using the generic `plot()` (`plot.ssm_df`) function on a model fit object. Options exist to plot fitted or predicted values along with observations as either paired, 1-D time series (using the `type = 1` argument) or as 2-D tracks (`type = 2`) with 95% confidence intervals or ellipses. These plots provide a rapid check on SSM fits to data. Additionally, when the fitted SSM is the move persistence model (i.e. `model = 'mp'`), 1-D time-series (`type = 3`) or 2-D track plots (`type = 4`) of move persistence estimates can be displayed (Figure 1).

`fit_ssm()` model fits can be mapped using `map()` for single or multiple individuals. By default, `map()` uses the coastline data from the `rnatuarearth` R package (South, 2022) at medium or high resolution

(Figure 1), but can also use tiled maps for finer scale detail, via the `rosm` (Dunnington, 2019) and `ggspatial` (Dunnington, 2021) R packages. Mapping aesthetics (e.g. plot symbols, sizes, colours, fills) can be customized via the `aes` argument and use of the `aes_1st()` function. See code in SI for examples. Further details on mapping tracks are here: <https://ianjensen.github.io/aniMotum/articles/Mapping.html>.

All aniMotum visualizations draw on the `ggplot2` R package (Wickham, 2016), with multipanel plots also using the `patchwork` R package (Pedersen, 2020), and generally can be modified through additive calls in the usual `ggplot2` manner. See code in SI for examples.

### 2.5 | Simulation—`Sim()`, `sim_fit()`, `sim_filter()`

Track simulation can be a helpful, yet informal, way of evaluating the degree to which statistical movement models capture essential features of animal movement data (Michelot et al., 2017). A variety of movement processes are simulated using `sim()`, including the `rw`, `crw` and `mp` process models, as well as `rw` and `crw` models with switching between multiple movement states that are not available in aniMotum's SSM's. `sim()` is intended as a general tool for random track simulation for a variety of purposes such as qualitative comparisons to real data, or evaluation of SSM estimation biases.

The `sim_fit()` function is intended for random track simulation in support of habitat usage and similar analyses. `sim_fit()` extracts movement parameters from an SSM fit object and simulates random tracks based on these parameters and with the same duration as the SSM-estimated track. `sim_fit()` has optional arguments that impose additional constraints on the simulated tracks that would be too computationally challenging to incorporate into the SSM. Whereas the SSM is constrained by the data, the simulated tracks are relatively unconstrained and so the following arguments allow the simulations to more closely approximate the real tracks.

The argument `cpf = TRUE` allows the simulation of central place foragers by ensuring that tracks start and end at approximately the same location. Movements can also be constrained to remain mostly in water via a potential function (Preisler et al., 2013), using gradient rasters included in aniMotum and the `grad` and `beta` arguments. These are illustrated in the code for Application 3.3.

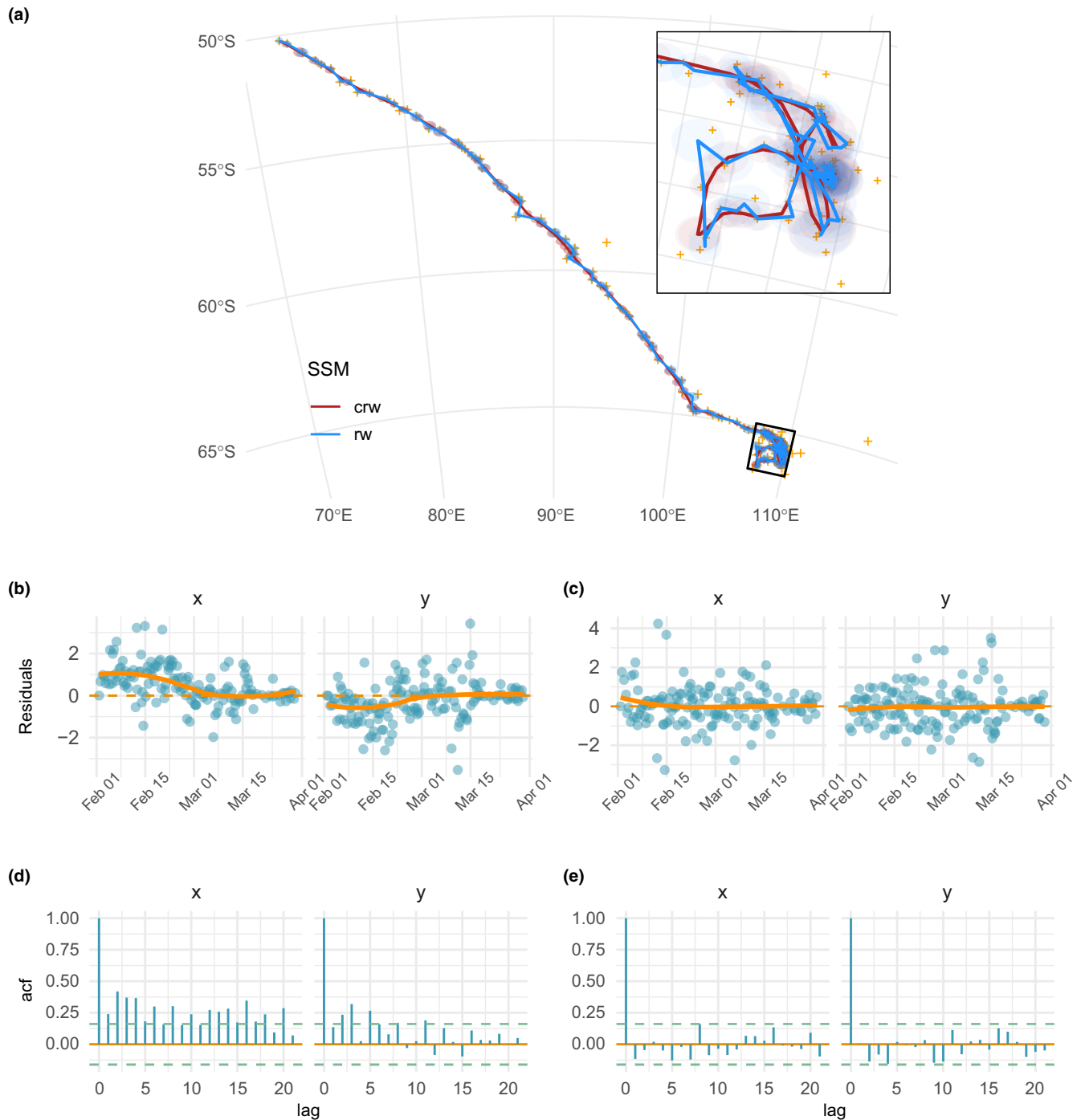
The `sim_filter()` function provides a tool to filter the simulated tracks based on their similarity to the SSM-estimated track. Here, similarity is calculated as the sum of normalized difference in net displacement (km) and overall bearing (deg) between the SSM-estimated track and the simulated tracks. Filtering uses one of two metrics that capture the differences between the two tracks (see `?similarity_flag` for more detail). These metrics are motivated by the 'flag value' described in Hazen et al. (2017). The quantile of flag values to be retained is specified via the `keep` argument; that is, `keep = 0.25` (the default) will return a `sim_fit` object containing those simulated tracks with flag values in the top 25% of values calculated for the input `sim_fit` object (Figure 1).

Further details on track simulation are here: [https://ianjensen.github.io/aniMotum/articles/Track\\_simulation.html](https://ianjensen.github.io/aniMotum/articles/Track_simulation.html).

## 2.6 | Path rerouting—route\_path()

As the SSMs implemented in aniMotum have no information about potential barriers to animal movement, it is possible for locations to be estimated in implausible locations; for example, on land for marine species. To overcome this, aniMotum makes use of the pathroutr R package (London, 2020) to efficiently re-route locations from land back to water by using visibility graphs (Jan

et al., 2014). route\_path() can be applied to either an SSM fit object or the simulations generated by sim\_fit(). Users must apply route\_path() themselves, this is not done automatically in aniMotum. When the input is an SSM fit object, the re-routed path can be appended to the object for visualization and use in subsequent analyses. When the input is a sim\_fit object, the locations within the simulation are replaced with the re-routed paths (Figure 1). We illustrate the latter in Application 3.3. Further details on path



**FIGURE 2** State-space model fits to a southern elephant seal track with finer scale detail inset (a), and diagnostic plots for assessing goodness of fit of the rw (b—prediction residual time series; d—prediction residual autocorrelation) and crw (c, e) state-space models. All residual plots generated using the plot.osar function. Processing time: 2.5 s.



rerouting are here: [https://ianjensen.github.io/aniMotum/articles/Path\\_rerouting.html](https://ianjensen.github.io/aniMotum/articles/Path_rerouting.html).

### 3 | APPLICATIONS

We illustrate the main capabilities of aniMotum with three applications. Complete R code for reproducing the applications and for gaining a deeper understanding of aniMotum functions is provided in the Supporting Information (Appendices S1–S4). Data used in the applications are available at <https://doi.org/10.5061/dryad.qz612jmjw>. Processing times for model fits and simulations in the applications are reported in their figure captions. All analyses conducted on a 2021 Apple MacBook Pro with an M1 Pro processor, 32 Gb Ram and R 4.3.

#### 3.1 | SSM validation with prediction residuals

We use a sub-adult male southern elephant seal *Mirounga leonina* track included as example data in aniMotum (sese2, id: ct36-E-09), sourced from the Australian Integrated Marine Observing System (IMOS; data publicly available via [imos.aodn.org.au](https://imos.aodn.org.au)) deployments at Iles Kerguelen. The data are temporally irregular Argos Least-Squares locations. We fitted both the rw and crw models using `fit_ssm()` with a speed filter threshold (`vmax`) of  $4 \text{ ms}^{-1}$  and a 12-h time step. We

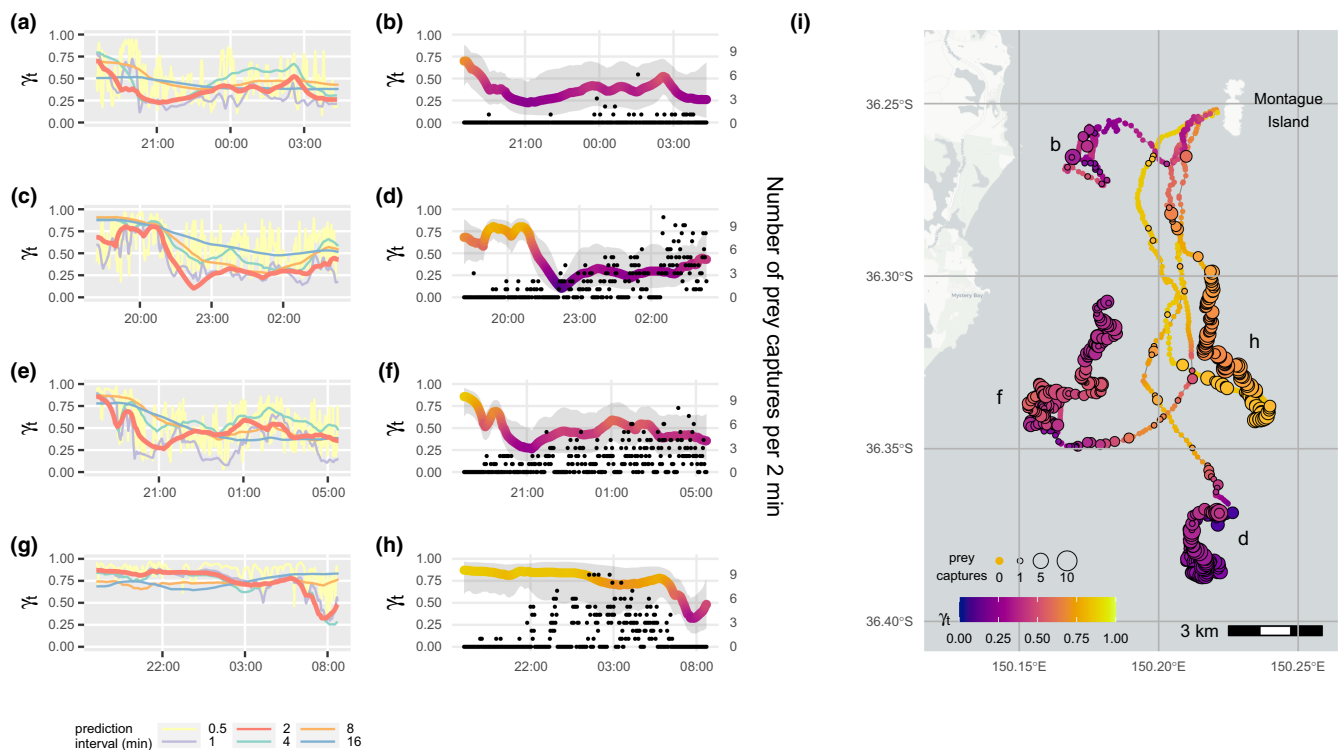
calculated prediction residuals using `osar()`, and then plotted the `osar` residuals to assess and compare the model fits (Figure 2).

Plots of predicted states with the observations suggest both models yield similar global fits (Figure 2a; compare blue vs. red lines and 95% confidence ellipses), but discrepancies are visible at a finer scale (Figure 2a, inset). Predicted locations from the two models differ by a median 6.62 km (range: 0.02, 53.02 km), and there are marked trends in the residuals for the rw model fit (Figure 2b) with significant positive autocorrelation in both the x and y directions (Figure 2d). The crw prediction residuals have no temporal trend and little autocorrelation (Figure 2c,e), implying the crw process model is a better fit to the data.

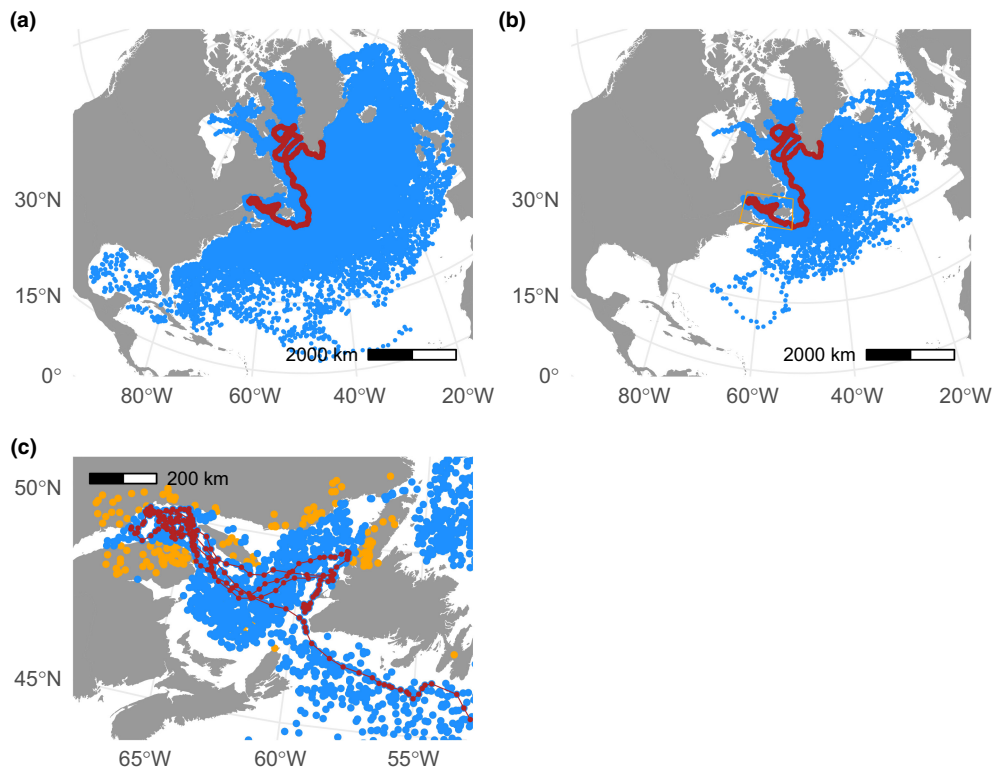
Although the crw process model provides a better fit, confidence ellipses around fitted and predicted location estimates should be examined. The `plot.ssm_df` function allows estimated location uncertainty to be visualized as 95% confidence intervals/ellipses in one (`type = 1`) or two dimensions (`type = 2`). When confidence ellipses are large, use of the location estimates without acknowledging their uncertainty in subsequent analyses may lead to biased results.

#### 3.2 | Inferring move persistence

To illustrate how move persistence ( $\gamma_t$ ) can be estimated from GPS data, we use four little penguin *Eudyptula minor* tracks of daily



**FIGURE 3** Inferred move persistence,  $\gamma_t$ , from time-regularized little penguin GPS tracks.  $\gamma_t$  estimates were compared across a range of prediction intervals (0.5–16 min; a, c, e, g; 2 min in bold orange). The 2-min  $\gamma_t$  estimates for each penguin are overlaid with the number of prey capture events, inferred from accelerometry data (black circles in b, d, f, h; grey envelopes are 95% CI's) and along SSM-predicted penguin tracks (i). Location symbol sizes in (i) are proportional to the number of prey capture events at 2-min intervals corresponding to the SSM predicted location times. Processing time: 4.091 min.



**FIGURE 4** Simulating (a) 100 movement paths from a correlated random walk process model; (b) filtering those tracks to select the top 25% based on their similarity to the original SSM-predicted track (red); and (c) re-routing simulated locations on land (orange) back to ocean (blue). The orange box in (b) indicates region magnified in (c). SSM-predicted track (red) overlaid in all panels for context. Processing time: 2.18 min.

foraging trips during the chick-rearing period from Montague Island, NSW, Australia (described in Phillips et al., 2022). The data are temporally irregular GPS locations, with high-frequency sampling (mean 15 s) intermittently disrupted by the birds' diving behaviour, and are assumed to have minimal measurement error. We fitted the crw SSM to the GPS data to predict locations at regular 2-min intervals, and assumed consistently small bivariate normal location measurement errors (i.e.  $\pm 30$  m SD). We then used `fit_mpm` to estimate  $\gamma_t$  from these regularized locations. As  $\gamma_t$  estimates are sensitive to time interval choice, we compare 2-min estimates with those from both shorter and longer prediction intervals. Finally, we visually compared  $\gamma_t$  estimates with prey capture events inferred from accelerometry data (see Carroll et al., 2014, for details) to ascertain whether reduced  $\gamma_t$  can indicate active foraging by these penguins. We provide an extra application to show how move persistence can be inferred from Argos data using `fit_ssm` (Appendix S3).

As  $\gamma_t$  is sensitive to choice of prediction time interval, it is important that a range of intervals be evaluated. Although the chosen 2-min prediction interval is longer than the GPS sampling frequency (15 s), it adequately captures the main changes in  $\gamma_t$  (Figure 3a,c,e,g). Shorter intervals (0.5 and 1 min) have substantial short-term variability that may not reflect real changes in movement pattern, while longer intervals (8 and 16 min) likely smooth out too much real variability in the movements.

The little penguin GPS tracks did not exhibit strong contrast in move persistence ( $\gamma_t$  range = 0.102, 0.871; Figure 3b,d,f,h).

Nonetheless, the move persistence estimates highlight changes in movement pattern over the course of the penguins' daily foraging trips (Figure 3i). The penguins departed Montague Is. with relatively fast movements, three directed southwards (Figure 3 tracks d, f, h in i) and one less directed and remaining closer to the island (track b), before slowing down and engaging in meandering movements (orange–purple in Figure 3i). The high  $\gamma_t$  estimates at the start of the penguins' foraging trips were associated with few prey captures (Figure 3b,d,f). However, periods of lower  $\gamma_t$  were only loosely associated with higher prey captures. For example, penguin 'h' had no prey captures as  $\gamma_t$  declined towards the end of its foraging trip (Figure 3h). This period of lower  $\gamma_t$  is likely indicative of resting as returning penguins usually wait offshore until after dusk before hauling out on the island. These results highlight that while move persistence can objectively indicate where animals spend more (low  $\gamma_t$ ) or less (high  $\gamma_t$ ) time, the behaviour(s) responsible for these patterns may not be obvious or consistent (e.g. Riaz et al., 2021).

### 3.3 | Simulating tracks from aniMotum model fits

To illustrate how to simulate tracks from aniMotum model fits, we use a juvenile harp seal *Pagophilus groenlandicus* tracked from the Gulf of St Lawrence, Canada (described in Grecian et al., 2022). The data are temporally irregular Argos locations including error ellipse



information. We fit the crw model using `fit_ssm` with a  $4 \text{ ms}^{-1}$  speed filter threshold (`vmax`) and a 12-h prediction interval (`time.step`).

We simulate 50 animal movement paths from the crw process model using `sim_fit` and apply a potential function using the `grad` and `beta` arguments to constrain the simulated paths to largely remain in water. These tracks are then filtered based on their similarity to the original path using `sim_filter` and the top 25% retained (`keep = 0.25`) (Figure 4a,b). As the potential function does not guarantee all locations remain off land, we re-route any remaining simulated locations from land back to water using `route_path` (Figure 4c). In combination, these functions provide a pragmatic, non-statistical method to generate and objectively filter pseudo-tracks for use in movement or habitat modelling applications.

## 4 | CONCLUSIONS

The aniMotum package was developed to ease fitting state-space models for quality control of animal location data and for inference of behavioural change along animal tracks. We achieve these primarily through a simple yet extensible workflow, simple movement models and computational speed. Combined, these traits accommodate both novice or occasional and advanced users, and facilitate use in automated, operational quality assurance/quality control processes for animal-borne sensor data (e.g. the AniBOS network; McMahon et al., 2021).

The aniMotum package is an intermediate analysis toolbox where location quality control typically occurs after some initial data processing but prior to any comprehensive, final analysis. Similarly, the move persistence models rapidly identify changes in movement behaviour along animal tracks without any required a priori knowledge about the kind or number of behavioural states potentially hidden within the data. Subsequent, detailed behavioural analysis could entail use of a hidden Markov model to infer potential drivers of observed movements (e.g. using the 'moveHMM' or 'momentuHMM' R packages; Michelot et al., 2016; McClintock & Michelot, 2018). Alternatively, move persistence-covariate relationships among multiple individuals can be inferred rapidly in a mixed-effects model framework using the `mpmm` R package (Jonsen et al., 2019, <https://github.com/ianjonsen/mpmm>).

The aniMotum track simulation tools support habitat usage analyses, allowing habitat availability (Aarts et al., 2012) to be estimated by generating 'background' points representing a null model of the distribution of foraging animals in the absence of external drivers (Raymond et al., 2015). As the generation of such null locations can impact the performance of habitat usage models (Phillips et al., 2009), aniMotum allows simulated tracks to be filtered based on their similarity to the SSM-predicted tracks. Eventually, aniMotum will allow simulation from the posterior of SSM-predicted tracks, whereby the simulated tracks are constrained by the estimated locations and their uncertainty (e.g. the 'crawl' R package; Johnson et al., 2008).

## 4.1 | The future of aniMotum

The current aniMotum version (v 1.1) is available from the R-universe repository (<https://ianjonsen.r-universe.dev/ui#package:aniMotum>), and can be installed from source within R via `install.packages("aniMotum", repos = "https://ianjonsen.r-universe.dev")`. Other installation options exist and we refer users to additional information here (<https://github.com/ianjonsen/aniMotum>).

The aniMotum package will undergo further development, expanding the range of available SSMs via new movement process models and/or enhanced measurement models for other electronic tracking data types. Additionally, we will seek to enhance integration with other R packages for processing and analysis of animal movement data, where feasible. Feedback from users is invaluable and encouraged. Users may submit bug reports and enhancement suggestions via the aniMotum GitHub issues page (<https://github.com/ianjonsen/aniMotum/issues/new/choose>). More general feedback is always welcome by contacting the lead author directly.

### AUTHOR CONTRIBUTIONS

Ian Jonsen developed the R package; James Grecian contributed harp seal data and to the R package; Lachlan Phillips, Gemma Carroll and Rob Harcourt contributed little penguin data; Clive McMahon, Rob Harcourt and Mark Hindell contributed Southern elephant seal data; Ian Jonsen and Toby Patterson developed the state-space models; Ian Jonsen wrote an initial draft of the manuscript with contributions from James Grecian; all authors contributed to the final manuscript.

### ACKNOWLEDGEMENTS

I.D.J. 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. T.A.P. was supported by CSIRO Oceans & Atmosphere internal research funding scheme. The Integrated Marine Observing System (IMOS) supported seal fieldwork. IMOS is a national collaborative research infrastructure, supported by the Australian Government and operated by a consortium of institutions as an unincorporated joint venture, with the University of Tasmania as Lead Agent. Fieldwork at Illes Kerguelen was conducted as part of the IPEV programme no. 109 (PI H. WEIMERSKIRCH) and of the SNO-MEMO programme (PI C. GUINET) in collaboration with IMOS. CTD tags were partly funded by CNES-TOSCA and IMOS. Little penguin fieldwork was supported by an Australian Research Council Linkage grant to I.D.J., G.C. and R.G.H. (LP160100162). Harp seal fieldwork was funded by Fisheries and Oceans Canada, a bursary from Department for Business, Energy and Industrial Strategy (BEIS) administered by the NERC Arctic Office, and the ARISE project (NE/P006035/1 and NE/P00623X/1), part of the Changing Arctic Ocean programme

jointly funded by the UKRI Natural Environment Research Council (NERC) and the German Federal Ministry of Education and Research (BMBF). Animal research protocols were approved by the University of Tasmania Animal Ethics Committee (A12141, A14523) and the Comité d'éthique Anses/ENVA/UPEC (APAFIS: 21375) (southern elephant seals); Macquarie University Animal Ethics Committee (ARA 2014/057-17; little penguins); and the Canadian Council on Animal Care (NAFC2017-2; harp seal). Open access publishing facilitated by Macquarie University, as part of the Wiley - Macquarie University agreement via the Council of Australian University Librarians. Open access publishing facilitated by Macquarie University, as part of the Wiley - Macquarie University agreement via the Council of Australian University Librarians.

## CONFLICT OF INTEREST

All authors declare no conflicts of interest.

## PEER REVIEW

The peer review history for this article is available at <https://www.webofscience.com/api/gateway/wos/peer-review/10.1111/2041-210X.14060>.

## DATA AVAILABILITY STATEMENT

All code used here is provided in the aniMotum package for R or in the Supporting Information. Machine-readable versions of the R code used in the Applications are available on GitHub (<https://github.com/ianjensen/foieGras.paper/tree/main/Sl>). The data have been archived in the Dryad Digital Repository: <https://doi.org/10.5061/dryad.qz612jmjw>. The R package aniMotum (v 1.1) is archived in the Zenodo repository: (<https://doi.org/10.5281/zenodo.7425388>).

## ORCID

Ian D. Jonsen  <https://orcid.org/0000-0001-5423-6076>  
 W. James Grecian  <https://orcid.org/0000-0002-6428-719X>  
 Lachlan Phillips  <https://orcid.org/0000-0002-7635-2817>  
 Gemma Carroll  <https://orcid.org/0000-0001-7776-0946>  
 Clive McMahon  <https://orcid.org/0000-0001-5241-8917>  
 Robert G. Harcourt  <https://orcid.org/0000-0003-4666-2934>  
 Mark A. Hindell  <https://orcid.org/0000-0002-7823-7185>  
 Toby A. Patterson  <https://orcid.org/0000-0002-7150-9205>

## REFERENCES

- Aarts, G., Fieberg, J., & Matthiopoulos, J. (2012). Comparative interpretation of count, presence-absence and point methods for species distribution models. *Methods in Ecology and Evolution*, 3, 177–187.
- Calabrese, J. M., Fleming, C. H., & Gurarie, E. (2016). Ctmm: An R package for analyzing animal relocation data as a continuous-time stochastic process. *Methods in Ecology and Evolution*, 7, 1124–1132.
- Carroll, G., Slip, D., Jonsen, I., & Harcourt, R. (2014). Supervised accelerometer analysis can identify prey capture by penguins at sea. *Journal of Experimental Biology*, 217, 4295–4302.
- Dunnington, D. (2019). *Rosm: Plot raster map tiles from open street map and other sources*. R Package Version 0.2.5.
- Dunnington, D. (2021). *Ggsatial: Spatial data framework for ggplot2*. R Package Version 1.1.5.
- Grecian, W. J., Stenson, G. B., Biuw, M., Boehme, L., Folkow, L. P., Goulet, P. J., Jonsen, I. D., Malde, A., Nordøy, E. S., Rosing-Asvid, A., & Smout, S. (2022). Environmental drivers of population-level variation in the migratory and diving ontogeny of an arctic top predator. *Royal Society Open Science*, 9, 211042.
- Hazen, E. L., Palacios, D. M., Forney, K. A., Howell, E. A., Becker, E., Hoover, A. L., Irvine, L., DeAngelis, M., Bograd, S. J., Mate, B. R., & Bailey, H. (2017). WhaleWatch: A dynamic management tool for predicting blue whale density in the California current. *Journal of Applied Ecology*, 54, 1415–1428.
- Jan, G. E., Sun, C. C., Tsai, W. C., & Lin, T. H. (2014). An  $O(n \log n)$  shortest path algorithm based on delaunay triangulation. *IEEE/ASME Transactions on Mechatronics*, 19, 660–666.
- Johnson, D. S., London, J. M., Lea, M., & Durban, J. W. (2008). Continuous-time correlated random walk model for animal telemetry data. *Ecology*, 89, 1208–1215.
- Jonsen, I., Flemming, J., & Myers, R. (2005). Robust state-space modeling of animal movement data. *Ecology*, 86, 2874–2880.
- Jonsen, I. D., 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.
- 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.
- Kays, R., Crofoot, M. C., Jetz, W., & Wikelski, M. (2015). Terrestrial animal tracking as an eye on life and planet. *Science*, 348, aaa2478.
- 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.
- London, J. M. (2020). *Pathroul: An R package for (Re-)routing paths around barriers*.
- 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.
- McMahon, C. R., Roquet, F., Baudel, S., Belbeoch, M., Bestley, S., Blight, C., Boehme, L., Carse, F., Costa, D. P., Fedak, M. A., Guinet, C., Harcourt, R., Heslop, E., Hindell, M. A., Hoenner, X., Holland, K., Holland, M., Jaine, F. R. A., Jeanniard du Dot, T., ... Woodward, B. (2021). Animal borne ocean sensors—Anibos—An essential component of the global ocean observing system. *Frontiers in Marine Science*, 8, 751840.
- Michelot, T., Langrock, R., Bestley, S., Jonsen, I. D., Photopoulou, T., & Patterson, T. A. (2017). Estimation and simulation of foraging trips in land-based marine predators. *Ecology*, 98, 1932–1944.
- 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.
- Patterson, T. A., Thomas, L., Wilcox, C., Ovaskainen, O., & Matthiopoulos, J. (2008). State-space models of individual animal movement. *Trends in Ecology & Evolution*, 23, 87–94.
- Pedersen, T. L. (2020). *Patchwork: The composer of plots*. R package version 1.1.1.
- Phillips, L., Carroll, G., Jonsen, I., Harcourt, R., Brierley, A. S., Wilkins, A., & Cox, M. (2022). Variability in prey field structure drives inter-annual differences in prey encounter by a marine predator, the little penguin. *Royal Society Open Science*, 9, 220028.
- Phillips, S. J., Dudík, M., Elith, J., Graham, C. H., Lehmann, A., Leathwick, J., & Ferrier, S. (2009). Sample selection bias and presence-only

- distribution models: Implications for background and pseudo-absence data. *Ecological Applications*, 19, 181–197.
- Preisler, H. K., Ager, A. A., & Wisdom, M. J. (2013). Analyzing animal movement patterns using potential functions. *Ecosphere*, 4, 1–13.
- R Core Team. (2021). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing.
- Raymond, B., Lea, M. A., Patterson, T., Andrews-Goff, V., Sharples, R., Charrassin, J. B., Cottin, M., Emmerson, L., Gales, N., Gales, R., Goldsworthy, S. D., Harcourt, R., Kato, A., Kirkwood, R., Lawton, K., Ropert-Coudert, Y., Southwell, C., van den Hoff, J., Wienecke, B., ... Hindell, M. A. (2015). Important marine habitat off East Antarctica revealed by two decades of multi-species predator tracking. *Ecography*, 38, 121–129.
- Riaz, J., Bestley, S., Wotherspoon, S., & Emmerson, L. (2021). Horizontal-vertical movement relationships: Adélie penguins forage continuously throughout provisioning trips. *Movement Ecology*, 9, 43.
- South, A. (2022). *Rnaturalearth: World map data from natural earth*. R package version 0.3.0.
- 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*, 24, 317–339.
- Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag.

## SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

**Appendix S1:** R code for Application 3.1.

**Appendix S2:** Code for Application 3.2.

**Appendix S3:** Inferring move persistence from southern elephant seal Argos tracking data.

**Appendix S4:** Code for Application 3.3.

**How to cite this article:** Jonsen, I. D., Grecian, W. J., Phillips, L., Carroll, G., McMahon, C., Harcourt, R. G., Hindell, M. A., & Patterson, T. A. (2023). aniMotum, an R package for animal movement data: Rapid quality control, behavioural estimation and simulation. *Methods in Ecology and Evolution*, 14, 806–816. <https://doi.org/10.1111/2041-210X.14060>