Received: 20 September 2024 | Accepted: 4 March 2025

DOI: 10.1111/2041-210X.70029

BRITISH ECOLOGICAL SOCIETY Methods in Ecology and Evolution

APPLICATION

patter: Particle algorithms for animal tracking in R and Julia

Edward Lavender[1] | **Andreas Scheidegger**[1] | **Carlo Albert**[1] | **Stanisław W. Biber**[2] | **Janine Illian**[3] | **James Thorburn**[4,5] | **Sophie Smout**[6] | **Helen Moor**[1]

[1]Department of Systems Analysis, Integrated Assessment and Modelling, Eawag Swiss Federal Institute of Aquatic Science and Technology, Dübendorf, Switzerland; [2]School of Engineering Mathematics and Technology, University of Bristol, Bristol, UK; [3]School of Mathematics and Statistics, University of Glasgow, Glasgow, UK; [4]School of Applied Sciences, Edinburgh Napier University, Edinburgh, UK; [5]Centre for Conservation and Restoration Science, Edinburgh Napier University, Edinburgh, UK and [6]Scottish Oceans Institute, University of St Andrews, St Andrews, UK

Correspondence
Edward Lavender
Email: edward.lavender@eawag.ch

Funding information
Department of Systems Analysis, Integrated Assessment and Modelling, Eawag Swiss Federal Institute of Aquatic Science and Technology

Handling Editor: Aaron Ellison

Abstract

1. State-space models are a powerful modelling framework in movement ecology that represents individual movements and the processes connecting movements to observations. However, fitting state-space models to animal-tracking data can be difficult and computationally expensive.
2. Here, we introduce patter, a package that provides particle filtering and smoothing algorithms that fit Bayesian state-space models to tracking data, with a focus on data from aquatic animals in receiver arrays. patter is written in R, with a performant Julia backend. Package functionality supports data simulation, preparation, filtering, smoothing and mapping.
3. In two examples, we demonstrate how to implement patter to reconstruct the movements of a tagged animal in an acoustic telemetry system from acoustic detections and ancillary observations. With perfect information, the particle filter reconstructs the true (unobserved) movement path (Example One). More generally, particle algorithms represent an individual's possible location probabilistically as a weighted series of samples ('particles'). In our illustration, we resolve an individual's (unobserved) location every 2 min during 1 month and use particles to visualise movements, map space use and quantify residency (Example Two).
4. patter facilitates robust, flexible and efficient analyses of animal-tracking data. The methods are widely applicable and enable refined analyses of space use, home ranges and residency.

KEYWORDS

Bayesian inference, movement ecology, package, particle filter, passive acoustic telemetry, state-space model

1 | INTRODUCTION

The field of movement ecology has expanded in recent decades (Nathan et al., 2008; Rafiq et al., 2021). Electronic tagging and tracking technologies are used to track animals across the globe, providing a 'panoramic window' into their lives (Hussey et al., 2015). In aquatic environments, satellite tracking has reconstructed the migrations of air-breathing animals (Hays & Hawkes, 2018), archival geolocation has revealed the transoceanic movements of pelagic fish (Block et al., 2005) and passive acoustic telemetry arrays have been established to track acoustically tagged animals from local to continental scales (Matley et al., 2022). This accumulation of animal-tracking data is outpacing the development of modelling methods and software packages for analysis (Rafiq et al., 2021).

State-space models (SSMs) have emerged as a powerful modelling framework for animal tracking (Patterson et al., 2008). An SSM is a hierarchical representation of a process-observer system in which the evolution of an unobserved ('latent') state ($\boldsymbol{s}$) through time (t) is imperfectly observed, generating 'noisy' observations ($\boldsymbol{y}_t$). Discrete-time SSMs for animal-tracking data model the movement process $f(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1})$ by which an animal's state (typically, location) evolves through time ($t \in \{1, \ldots, T\}$) and the observation processes $f(\boldsymbol{y}_t \mid \boldsymbol{s}_t)$ connecting movements to observations. The SSM thus forms a formal statistical framework within which it is possible to estimate the unobserved states of an animal that are of interest, while accounting for movement properties (including speeds and barriers) and observation processes (such as detectability). However, fitting SSMs can be challenging and computationally expensive (Patterson et al., 2008).

Particle filters are flexible Monte Carlo algorithms used to fit state-space models (Doucet & Johansen, 2009). In an animal-tracking context, a particle filter approximates the distribution of possible locations of an individual with a set of weighted samples termed 'particles' (Lavender et al., 2025a). A movement model simulates possible locations of the individual (i.e. $\boldsymbol{s}_t \sim f(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1})$) and observation model(s) weight particles in line with the probability of the observations (i.e. $f(\boldsymbol{y}_t \mid \boldsymbol{s}_t)$). By resampling particles in line with the weights, we duplicate particles that are compatible with the data and eliminate incompatible particles. The result is an approximation of the distribution of the individual's location at each time step, given the preceding data (i.e. the partial marginal distribution, $f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:t})$). Particle smoothers and samplers are extensions that approximate the full marginal ($f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:T})$) and the joint ($f(\boldsymbol{s}_{1:T} \mid \boldsymbol{y}_{1:T})$) distributions, respectively (Doucet & Johansen, 2009). Compared to alternative SSM-fitting methods for animal-tracking data, advantages of particle algorithms include their flexibility, scalability and the ease with which they can be intuitively understood. In the ecological literature, a handful of particle filtering routines have been developed, including for fish geolocation over coarse spatial scales (Liu et al., 2019). However, existing routines are relatively specialised, computationally intensive and require user expertise.

Here, we introduce patter, a package that provides particle filtering and smoothing algorithms for animal-tracking data, motivated by our research in acoustic telemetry systems. patter is written in R (Lavender, 2024a) and integrates a Julia backend, Patter.jl (Lavender, 2024b). Julia is a programming language that combines the ease of use of an interpreted language like R with the speed of a compiled language like C++ (Bezanson et al., 2017). patter includes routines for simulation, data preparation, particle filtering, smoothing and mapping. These routines extend the acoustic telemetry and animal-tracking package ecosystems (Joo et al., 2020; Kraft et al., 2023). In the context of passive acoustic telemetry, patter is unique in the provision of routines that reconstruct individual movements and patterns of space use within a coherent probabilistic framework. The routines enable refined analyses of space use, home ranges and residency.

2 | METHODOLOGY

2.1 | Model formulation

The statistical methodology is described in Lavender et al. (2025a). This section provides a summary.

2.1.1 | Posterior

We consider a Bayesian state-space model for the state of a tagged animal; that is, the joint distribution $f(\boldsymbol{s}_{1:T} \mid \boldsymbol{y}_{1:T})$, where $\boldsymbol{s}_t = (s_x, s_y)$ denotes the state (a two-dimensional location in our examples), $\boldsymbol{y}_t$ denotes observations and $t \in \{1, 2, \ldots, T\}$ indexes time steps. The joint distribution is proportional to the product of a prior (the movement process) and the likelihood (the observation process), that is, $f(\boldsymbol{s}_{1:T} \mid \boldsymbol{y}_{1:T}) \propto f(\boldsymbol{s}_{t=1}) f(\boldsymbol{y}_{t=1} \mid \boldsymbol{s}_{t=1}) \prod_{t=2}^{T} f(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1}) f(\boldsymbol{y}_t \mid \boldsymbol{s}_t)$.

2.1.2 | Prior

The prior comprises a probability density distribution of the animal's initial location ($f(\boldsymbol{s}_{t=1})$) and a movement model ($f(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1})$). A simple model for $f(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1})$ in the two-dimensional case is a discrete-time random walk, where $\boldsymbol{s}_t = (s_{x,t-1} + d_t \cos \phi_t, s_{y,t-1} + d_t \sin \phi_t)$ and d (step length) and ϕ (heading) are independently distributed random variables, restricted by boundary conditions (e.g. land).

2.1.3 | Likelihood

The likelihood measures the probability of the observations given the latent states ($\boldsymbol{s}_t$). In an acoustic telemetry system, observations may include acoustic measurements at each of M receivers (an $M \times T$ matrix, $\boldsymbol{y}^{(A)}$) and ancillary measurements, such as depths (a row vector, $\boldsymbol{y}^{(D)}$). By way of example, here we consider a combined dataset $\boldsymbol{y} = \{\boldsymbol{y}^{(A)}, \boldsymbol{y}^{(D)}\}$ and the likelihood $f(\boldsymbol{y}_t \mid \boldsymbol{s}_t) = f\left(y_t^{(A)} \middle| \boldsymbol{s}_t\right) f\left(y_t^{(D)} \middle| \boldsymbol{s}_t\right)$.

2.1.4 | Acoustic measurements

The likelihood of the acoustic measurements at time t ($\mathbf{y}_t^{(A)}$), which comprise detections (1) or non-detections (0) at each operational receiver (i.e. $y_{k,t}^{(A)} \in \{0,1\}$), can be modelled using the Bernoulli probability mass function, $f\left(\mathbf{y}_t^{(A)} \middle| \mathbf{s}_t\right) = \prod_k p_{k,t}(\mathbf{s}_t)^{y_{k,t}^{(A)}}\left(1-p_{k,t}(\mathbf{s}_t)\right)^{1-y_{k,t}^{(A)}}$ (assuming independence). We typically model detection probability, p, as a function of the distance between the receiver (at location $\mathbf{r}_k$) and transmitter (at $\mathbf{s}$), such as $p_{k,t}(\mathbf{s}_t) = \begin{cases} \left(1+\mathrm{e}^{-(\alpha-\beta\times\|\mathbf{s}_t-\mathbf{r}_k\|)}\right)^{-1} & \text{if } \|\mathbf{s}_t-\mathbf{r}_k\|<\gamma \\ 0 & \text{otherwise} \end{cases}$, where α and β are parameters and γ is the detection range.

2.1.5 | Depth measurements

A simple model for the likelihood of a depth observation is:

$$f\left(y_t^{(D)} \middle| \mathbf{s}_t\right) = \begin{cases} z_t & \text{if } b(\mathbf{s}_t) - \epsilon_{\text{shallow}}(\mathbf{s}_t) \le y_t^{(D)} \le b(\mathbf{s}_t) + \epsilon_{\text{deep}}(\mathbf{s}_t) \\ 0 & \text{otherwise} \end{cases},$$

where $z_t = \left(\epsilon_{\text{deep}}(\mathbf{s}_t) + \epsilon_{\text{shallow}}(\mathbf{s}_t)\right)^{-1}$. This requires $y_t^{(D)}$ to be within a window around the bathymetric depth $b(\mathbf{s}_t)$. The window's width is defined by the shallow and deep depth-adjustment functions, $\epsilon_{\text{shallow}}(\mathbf{s}_t)$, $\epsilon_{\text{deep}}(\mathbf{s}_t) \le b(\mathbf{s}_t)$. These functions capture observational uncertainty (i.e. tag accuracy and the accuracy of the bathymetric measurement) and can be tailored for different species (depending on how much time they spend near the seabed). For benthic species, small errors ($\epsilon_{\text{shallow}}(\mathbf{s}_t), \epsilon_{\text{deep}}(\mathbf{s}_t) \ll b(\mathbf{s}_t)$) only permit observations close to the seabed; for pelagic species, larger $\epsilon_{\text{shallow}}(\mathbf{s}_t)$ values permit observations in the water column.

2.2 | Inference

2.2.1 | Filter

For inference, we begin with the partial marginal distribution, $f(\mathbf{s}_t \mid \mathbf{y}_{1:t})$. Particle filters approximate $f(\mathbf{s}_t \mid \mathbf{y}_{1:t})$ as a sum of N weighted particles, that is, $f(\mathbf{s}_t \mid \mathbf{y}_{1:t}) \approx \sum_{i=1}^{N} \delta(\mathbf{s}_t - \mathbf{s}_{i,t})w_i$, where δ is the Dirac delta function, w denotes normalised weights and i indexes particles. We perform inference for the latent states, assuming static parameters (in the movement and observation models) can be specified using supporting datasets, domain expertise and literature. Starting with an initial set of particles sampled from the prior (i.e. $\mathbf{s}_{i,1} \sim f(\mathbf{s}_1)$), the filter iteratively simulates particles via the movement model (i.e. $\mathbf{s}_{i,t} \sim f(\mathbf{s}_{i,t} \mid \mathbf{s}_{i,t-1})$), weights particles in line with the likelihood (via $w_{i,t} \propto w_{i,t-1} f(\mathbf{y}_t \mid \mathbf{s}_{i,t})$) and resamples particles accordingly.

2.2.2 | Smoother

Particle smoothers re-weight filtered particles to approximate the full marginal, $f(\mathbf{s}_t \mid \mathbf{y}_{1:T})$. The two-filter smoother uses N particles ($\mathbf{s}_{i,t}$) from a forward filter (with weights $w_{i,t}$) and N particles ($\tilde{\mathbf{s}}_{j,t}$) from a backward filter (with weights $\tilde{w}_{t,j}$). The distribution $f(\mathbf{s}_t \mid \mathbf{y}_{1:T})$ is approximated as a sum of re-weighted particles via $f(\mathbf{s}_t \mid \mathbf{y}_{1:T}) \approx \sum_{j=1}^{N} \delta(\mathbf{s}_t - \tilde{\mathbf{s}}_{j,t})\tilde{w}_{j,t|T}$, where the smoothed weights for each particle $\tilde{\mathbf{s}}_{j,t}$ represent a summation over all possible movements from preceding particles on the forward filter, that is, $\tilde{w}_{j,t|T} \approx \tilde{w}_{j,t} \sum_{i=1}^{N} f(\tilde{\mathbf{s}}_{j,t} \mid \mathbf{s}_{i,t-1})w_{i,t-1}$. We use samples from $f(\mathbf{s}_t \mid \mathbf{y}_{1:T})$ to map utilisation distributions (Lavender et al., 2025a). Sampling from the joint distribution, $f(\mathbf{s}_{1:T} \mid \mathbf{y}_{1:T})$, is more expensive and beyond the scope of this contribution.

3 | PACKAGE

patter supports data input, algorithms and mapping (Figure 1).

For data input, patter provides sim_*() functions for de novo simulation or accepts real-world datasets.

Particle algorithms are implemented by pf_*() functions. Filtering and smoothing are implemented via pf_filter() and pf_smoother_two_filter(). We provide movement models and methods that evaluate the likelihood of acoustic and depth observations, but algorithm components can be customised as required. The main output is a data.table of particles. The core routines use Patter.jl. JuliaCall implements the coupling between the patter and Patter.jl packages (Li, 2019). Movement and observation models are multithreaded and designed for numerical stability. We anticipate that most users will prefer the R front end, but Patter.jl can also be used directly.

Mapping functions (map_*()) facilitate subsequent analysis, including the reconstruction of utilisation distributions (Lavender et al., 2025a). Routines include map_pou(), which maps probability-of-use across a grid; map_dens(), which incorporates kernel smoothing; and map_hr_*() functions, which compute home ranges.

4 | EXAMPLES

4.1 | Overview

We provide two examples using simulated data. Both examples consider the movements of a benthic animal in a hypothetical acoustic array spanning a marine protected area (MPA) in Scotland (Figure 2). The study area is defined by a 100 × 100 m resolution bathymetry grid. We base the grid on real-world data (Howe et al., 2014) but, for the purposes of our first example, add some random noise such that each cell's depth is unique. Within this region, we tag an animal with an acoustic transmitter and an archival depth tag. We imagine that both tags operate at a resolution of 2 min and simulate a discrete-time random walk at this resolution over a 1-month period (Figure 2b,c). We simulate acoustic and depth observations arising from the simulated path and apply our algorithms to these data to reconstruct movements and patterns of space use. In the first example,

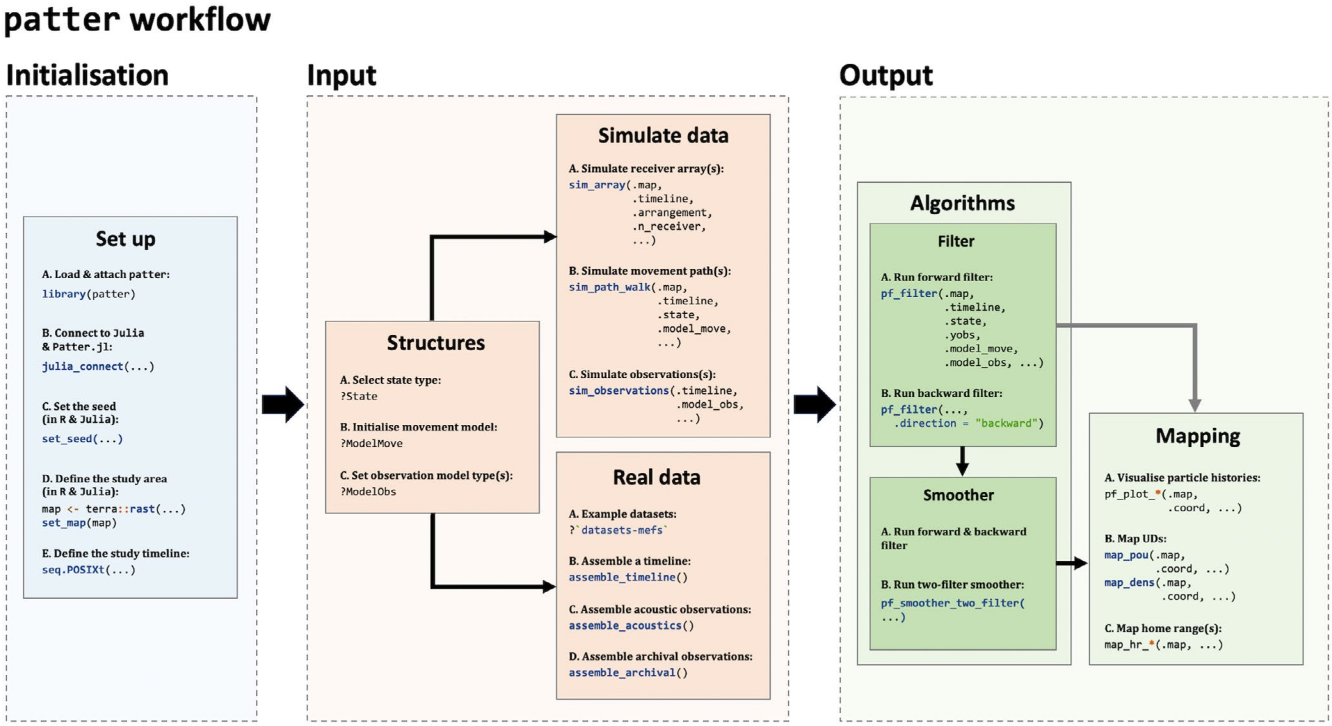

FIGURE 1 Package overview.

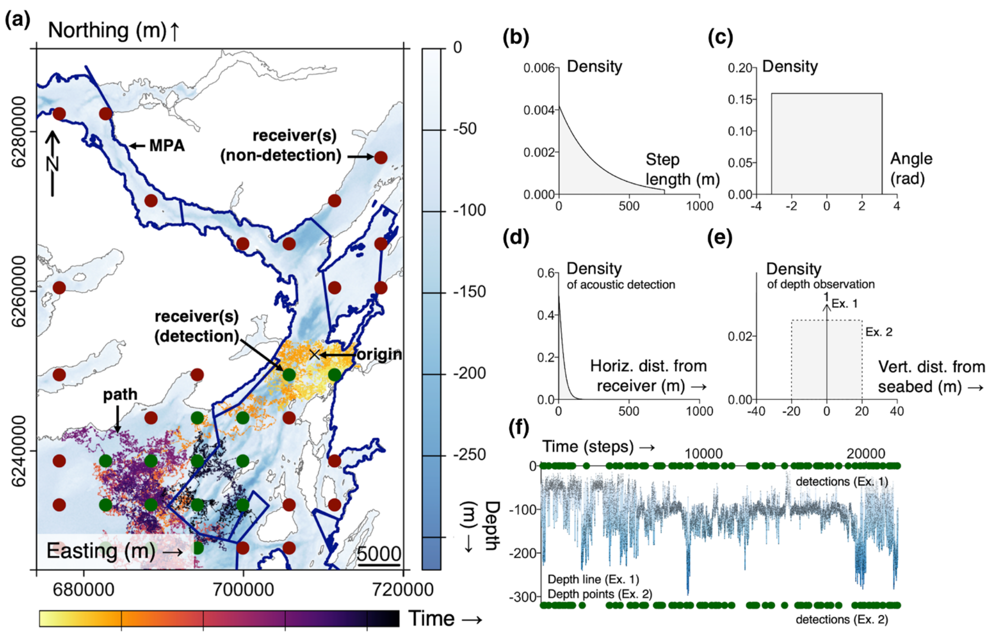

FIGURE 2 A state-space model for animal-tracking data. (a) A study area, including a simulated movement path (coloured by time) and acoustic receivers (sized by detection range and coloured by detection(s)/non-detection). (b, c) The components of the random walk used to simulate and model movements. (d, e) The observation models used to simulate and model acoustic and depth observations arising from the simulated path in the two worked examples; the acoustic observation model is constant, but the depth model differs (see Section 4.1). (f) The simulated time series for each example.

we simulate observations in such a way that the depth observation exactly defines the location of the individual (which is situated on the seafloor) (Figure 2d–f). This example will demonstrate that, in the absence of uncertainty, the particle filter reconstructs the true movement path. In the second example, we simulate observations probabilistically and demonstrate the representation of uncertainty in particle algorithms and maps of space use (Figure 2d–f). The following sections showcase key functions and arguments (Figure 1). Additional arguments are denoted by ellipses. Complete code is available online (Lavender et al., 2025b).

4.2 | Implementation

4.2.1 | Simulation

We begin with essential initiation:

```
library(patter)
library(data.table)
julia_connect()
```

Next, we define the study system:

```
# Define study period
timeline <- seq(as.POSIXct("2023-01-01 12:00:00", tz = "UTC"),
                as.POSIXct("2023-01-31 23:58:00", tz = "UTC"),
                by = "2 mins")

# Define study site (map: SpatRaster)
set_map(map)
```

We simulate an acoustic array (i.e. data.table of receivers). We include three receiver columns that represent observation model (detection probability) parameters:

```
moorings <- sim_array(map,
                      timeline,
                      .n_receiver = 100L,
                      .arrangement = "regular",
                      .receiver_alpha = 4,
                      .receiver_beta = -0.01,
                      .receiver_gamma = 750)
```

We then simulate a two-dimensional random walk in this area:

```
# Define state type for 2D walk
# (s_t = (s_{x, t}, s_{y, t}), truncated by boundaries)
state <- "StateXY"

# Define movement model that updates s_t
# d_t ~ TruncatedGamma(k, theta, 0, mobility)
# phi_t ~ Uniform(a, b)
model_move <-
  move_xy(dbn_length = "truncated(Gamma(1.0, 250.0), upper=
750.0)",
          dbn_angle = "Uniform(-pi, pi)")

# Simulate 2D path (data.table of states)
# s_t = (s_{x, t-1} + d_t * cos(phi_t), s_{y, t-1} + d_t * sin(phi_t)),
# subject to boundary conditions
path <- sim_path_walk(map,
                      timeline,
                      state,
                      model_move, ...)
```

Next, we simulate acoustic and depth observations. This requires defining a vector of observation model (ModelObs) structures (which hold model parameters) and a corresponding list of data.tables (with those parameters). For both examples, we simulate acoustic observations from a truncated logistic model (for which we provide the ModelObsAcousticLogisTrunc structure and the essential parameters are defined in moorings). To simulate depths, we consider a simple version of the uniform model described previously, as implemented by the ModelObsDepthUniform structure, where $\epsilon_{\text{shallow}}(\boldsymbol{s}_t)$ and $\epsilon_{\text{deep}}(\boldsymbol{s}_t)$ are the constants depth_shallow_eps and depth_deep_eps.

In the first example, we imagine an animal found exclusively on the seabed, the depth of which is known exactly, giving parameters:

```
data.table(sensor_id = 1L,
           depth_shallow_eps = 0,
           depth_deep_eps = 0)
```

In the second example, we incorporate uncertainty:

```
data.table(sensor_id = 1L,
           depth_shallow_eps = 20,
           depth_deep_eps = 20)
```

For each example, sim_observations() simulates a list of observations:

```
model_obs <- c("ModelObsAcousticLogisTrunc","ModelObsDepthUni-
form")
obs <- sim_observations(timeline,
                        model_obs,
                        .model_obs_pars = list(...))
```

where list(...) denotes the parameter data.tables for the relevant example.

4.2.2 | Particle filter

The particle filter is implemented via:

```
pf_filter(
  map,
  timeline,
  state,
  xinit,
  .yobs = list(...),
  .model_obs  = model_obs,
  .model_move = model_move,
  .n_particle = 1e5L,
  .direction  = "forward", ...
)
```

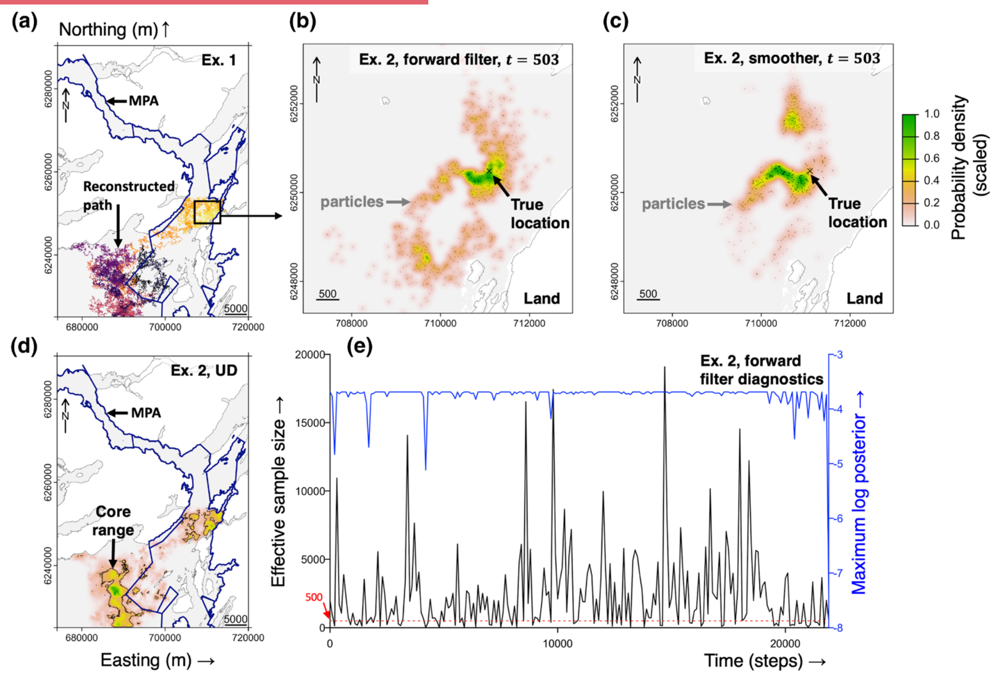

FIGURE 3 Outputs for the first (a) and second (b–e) worked examples. (a) is the movement path reconstructed by the particle filter for the first example. The simulated path is recovered perfectly (at grid resolution) because the observations exactly define the individual's location. (b, c) show particles and scaled probability densities from the forward filter and two-filter smoother, respectively, at a selected time step. (d) maps the pattern of space use over the entire time series using smoothed particles. Core ranges contain 50% of the probability mass volume. Estimated residency in the MPA is 37.0% (the true value is 36.3%). (e) shows diagnostics from the forward filter. The minimum value of each statistic is shown for every 100 time steps.

where .xinit (optional) is the simulated tagging location and .yobs is the list of datasets for the relevant example. This returns a pf_particles-class object that includes a data.table of particles and diagnostic statistics. In the first example, the filter reconstructs the true (unobserved) path. In the second example, we generate a 'cloud' of particles at each time step, for which we examine particle diagnostics and proceed to smoothing.

4.2.3 | Particle smoother

Particle smoothing is implemented using outputs from a forward and backward filter via

pf_smoother_two_filter(...)

For illustration, we reconstruct the utilisation distribution and home range from smoothed particles via map_dens() and map_hr(). We also compare time spent in the MPA, estimated from the proportion of particles inside the MPA, to the truth.

This workflow is highly customisable. patter provides customisable, built-in structures for selected movement models (such as random walks) and observation types (such as acoustic observations). User-defined structures, for complete customisation, are also supported. See the package documentation for details.

4.3 | Results

In our first example, the particle filter reconstructs the simulated movement path perfectly (Figure 3a). In the second example, in which observations were simulated with error, the particle filter represents the individual's possible locations at each time step with a series of weighted particles that approximate $f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:t})$ (Figure 3b). The particle smoother re-weights filtered particles, approximating $f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:T})$ (Figure 3c). Smoothed particles can be used to map patterns of space use, estimate home ranges and quantify residency (Figure 3d). The quality of the smoothing depends on the filter. In this case, filter diagnostics are adequate (Figure 3e). Total computation time ranged from 5 to 32 min for examples 1–2 on a 2023 MacBook Pro (Apple M2 Pro, 32 GB RAM, 12 CPUs).

5 | DISCUSSION

patter provides a robust, fast and accessible implementation of particle algorithms for animal tracking, especially with passive acoustic telemetry (Lavender et al., 2025a). These algorithms represent the movement and observation processes that generate observations, including movement capacity, barriers to movement and detection probability, within a biologically and statistically sound framework. The movement and observation models are fully customisable, making the routines applicable in many real-world settings. However, understanding the settings in which different methods are more or less useful remains an important research area (Lavender et al., 2025a).

The patter workflow involves formulating a state-space model for the state of a tagged animal and performing inference via particle filtering and smoothing. A biological challenge during this process is the formulation of the movement and observation submodels. Currently, patter requires users to parameterise submodels a priori. This is a trade-off between inferential flexibility and computational efficiency. We consider the primary purpose of positional animal-tracking (e.g. acoustic telemetry) studies to learn about patterns of space use, rather than locomotory capacity or observational processes (such as detection probability). Other approaches, such as accelerometry studies and drift tests of detection probability, can provide more detailed information on the latter (especially in sparse acoustic arrays). We encourage users to leverage these data, alongside domain knowledge and literature, to parameterise models (Lavender et al., 2025a). That being said, we recognise that joint inference may be desirable in situations where data are sufficient and this is a possible future development. Joint inference is expensive, but there are routines, such as Hamiltonian Monte Carlo, that offer potential in this regard (Albert et al., 2015).

A key feature of patter is speed. We achieve competitive speeds by focusing inference on individual states, targeting marginal (rather than joint) distributions and via a performant Julia backend. While formal benchmarks are lacking, simple comparisons are instructive. Run times for a Python particle filtering application estimating daily posterior distributions for demersal fish over a 1-year period are approximately 1 h (Liu et al., 2019). Other geolocation routines that fit hidden Markov models via likelihood approximation typically require hours or days to derive daily geolocation estimates over the same timeframe (Pedersen et al., 2008). Inference using Markov Chain Monte Carlo algorithms (which directly sample static parameters and trajectories) is also often expensive. For example, we observed that Hostetter and Royle (2020)'s state-space model for acoustic detections ($T = 150$) requires ≈15 h to fit with their JAGS code on a standard computer. With our particle filtering–smoothing algorithm, the estimation of latent locations in this situation is soluble in seconds. While run times are not directly comparable, it is encouraging to see patter achieving speeds sufficient to make particle algorithms serious candidates for real-world analyses.

For applied studies, we suggest the use of simulations to guide method implementation and interpretation. Particle filters can be sensitive to model parameters and tuning settings (such as particle number) and system-specific simulations can inform input arguments and quantify sensitivity (Lavender et al., 2025a). For an example real-world analysis, we direct the reader to Lavender, Scheidegger, Albert, Biber, Aleynik, et al. (2025), who analysed acoustic and archival data from a Critically Endangered elasmobranch to quantify patterns of space use and site affinity in a Scottish Marine Protected Area. There is much to be learnt from applications in other settings, and we welcome community feedback as these developments are exploited.

AUTHOR CONTRIBUTIONS

Edward Lavender conceived the study and developed the methodology with Andreas Scheidegger, Carlo Albert and Helen Moor (Principal Investigator). This was motivated by work on the Movement Ecology of Flapper Skate project established by James Thorburn and an earlier modelling study led principally by Edward Lavender, Stanisław Biber, Janine Illian and Sophie Smout (Principal Investigator). In the current study, Edward Lavender and Andreas Scheidegger wrote the code. Edward Lavender analysed the data and led the writing of the manuscript. All authors contributed to drafts and gave approval for publication.

ACKNOWLEDGEMENTS

EL was supported by a postdoctoral researcher position at Eawag, funded by the Department of Systems Analysis, Integrated Assessment and Modelling.

CONFLICT OF INTEREST STATEMENT

The 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.70029.

DATA AVAILABILITY STATEMENT

The patter and Patter.jl packages are available on GitHub (Lavender, 2024a, 2024b). Code is available via https://doi.org/10.5281/zenodo.12771850 (Lavender et al., 2025b).

ORCID

Edward Lavender https://orcid.org/0000-0002-8040-7489
Andreas Scheidegger https://orcid.org/0000-0003-2575-2172
Carlo Albert https://orcid.org/0000-0001-9160-9601
Stanisław W. Biber https://orcid.org/0000-0001-5157-2904
Janine Illian https://orcid.org/0000-0002-6130-2796
James Thorburn https://orcid.org/0000-0002-4392-1737
Sophie Smout https://orcid.org/0000-0002-5125-9827
Helen Moor https://orcid.org/0000-0002-1340-2039

REFERENCES

Albert, C., Ulzega, S., & Stoop, R. (2015). Boosting Bayesian parameter inference of nonlinear stochastic differential equation models by Hamiltonian scale separation. *Physical Review E*, *93*, 043313. https://doi.org/10.1103/PhysRevE.93.043313

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98. https://doi.org/10.1137/141000671

Block, B. A., Teo, S. L. H., Walli, A., Boustany, A., Stokesbury, M. J. W., Farwell, C. J., Weng, K. C., Dewar, H., & Williams, T. D. (2005). Electronic tagging and population structure of Atlantic bluefin tuna. *Nature*, *434*(7037), 1121–1127. https://doi.org/10.1038/nature03463

Doucet, A., & Johansen, A. (2009). A tutorial on particle filtering and smoothing: Fifteen years later. In D. Crisan & B. Rozovskii (Eds.), *The Oxford handbook of nonlinear filtering*. Oxford University Press.

Hays, G. C., & Hawkes, L. A. (2018). Satellite tracking sea turtles: Opportunities and challenges to address key questions. *Frontiers in Marine Science*, *5*, 432. https://doi.org/10.3389/fmars.2018.00432

Hostetter, N. J., & Royle, J. A. (2020). Movement-assisted localization from acoustic telemetry data. *Movement Ecology*, *8*(1), 15. https://doi.org/10.1186/s40462-020-00199-6

Howe, J. A., Anderton, R., Arosio, R., Dove, D., Bradwell, T., Crump, P., Cooper, R., & Cocuccio, A. (2014). The seabed geomorphology and geological structure of the firth of Lorn, western Scotland, UK, as revealed by multibeam echo-sounder survey. *Earth and Environmental Science Transactions of the Royal Society of Edinburgh*, *105*(4), 273–284. https://doi.org/10.1017/S1755691015000146

Hussey, N. E., Kessel, S. T., Aarestrup, K., Cooke, S. J., Cowley, P. D., Fisk, A. T., Harcourt, R. G., Holland, K. N., Iverson, S. J., Kocik, J. F., Mills Flemming, J. E., & Whoriskey, F. G. (2015). Aquatic animal telemetry: A panoramic window into the underwater world. *Science*, *348*(6240), 1255642. https://doi.org/10.1126/science.1255642

Joo, R., Boone, M. E., Clay, T. A., Patrick, S. C., Clusella-Trullas, S., & Basille, M. (2020). Navigating through the R packages for movement. *Journal of Animal Ecology*, *89*(1), 248–267. https://doi.org/10.1111/1365-2656.13116

Kraft, S., Gandra, M., Lennox, R. J., Mourier, J., Winkler, A. C., & Abecasis, D. (2023). Residency and space use estimation methods based on passive acoustic telemetry data. *Movement Ecology*, *11*(1), 12. https://doi.org/10.1186/s40462-022-00364-z

Lavender, E. (2024a). *{patter}: Particle algorithms for animal movement in R* (Version 1.0.0) [R]. https://github.com/edwardlavender/patter

Lavender, E. (2024b). *{Patter.jl}: Particle algorithms for animal movement in Julia* (Version 1.0.0) [Julia]. https://github.com/edwardlavender/Patter.jl

Lavender, E., Scheidegger, A., Albert, C., Biber, S., Aleynik, D., Dodd, J., Cole, G., Wright, P. J., James, M., Illian, J., Smout, S., Thorburn, J., & Moor, H. (2025). Animal tracking with particle algorithms for conservation. *bioRxiv*. https://doi.org/10.1101/2025.02.13.638042

Lavender, E., Scheidegger, A., Albert, C., Biber, S., Illian, J., Thorburn, J., Smout, S., & Moor, H. (2025a). Particle algorithms for animal movement modelling in receiver arrays. *Methods in Ecology and Evolution*. https://doi.org/10.1101/2024.09.16.613223

Lavender, E., Scheidegger, A., Albert, C., Biber, S., Illian, J., Thorburn, J., Smout, S., & Moor, H. (2025b). patter: Particle algorithms for animal tracking in R and Julia (R code) [dataset]. *Zenodo*. https://doi.org/10.5281/zenodo.12771850

Li, C. (2019). JuliaCall: An R package for seamless integration between R and Julia. *Journal of Open Source Software*, *4*(35), 1284. https://doi.org/10.21105/joss.01284

Liu, C., Cowles, G. W., Zemeckis, D. R., Fay, G., Le Bris, A., & Cadrin, S. X. (2019). A hardware-accelerated particle filter for the geolocation of demersal fishes. *Fisheries Research*, *213*, 160–171. https://doi.org/10.1016/j.fishres.2019.01.019

Matley, J. K., Klinard, N. V., Barbosa Martins, A. P., Aarestrup, K., Aspillaga, E., Cooke, S. J., Cowley, P. D., Heupel, M. R., Lowe, C. G., Lowerre-Barbieri, S. K., Mitamura, H., Moore, J.-S., Simpfendorfer, C. A., Stokesbury, M. J. W., Taylor, M. D., Thorstad, E. B., Vandergoot, C. S., & Fisk, A. T. (2022). Global trends in aquatic animal tracking with acoustic telemetry. *Trends in Ecology & Evolution*, *37*(1), 79–94. https://doi.org/10.1016/j.tree.2021.09.001

Nathan, R., Getz, W. M., Revilla, E., Holyoak, M., Kadmon, R., Saltz, D., & Smouse, P. E. (2008). A movement ecology paradigm for unifying organismal movement research. *Proceedings of the National Academy of Sciences of the United States of America*, *105*(49), 19052–19059. https://doi.org/10.1073/pnas.0800375105

Patterson, T. A., Thomas, L., Wilcox, C., Ovaskainen, O., & Matthiopoulos, J. (2008). State–space models of individual animal movement. *Trends in Ecology & Evolution*, *23*(2), 87–94. https://doi.org/10.1016/j.tree.2007.10.009

Pedersen, M. W., Righton, D., Thygesen, U. H., Andersen, K. H., & Madsen, H. (2008). Geolocation of North Sea cod (*Gadus morhua*) using hidden Markov models and behavioural switching. *Canadian Journal of Fisheries and Aquatic Sciences*, *65*(11), 2367–2377. https://doi.org/10.1139/F08-144

Rafiq, K., Pitcher, B. J., Cornelsen, K., Hansen, K. W., King, A. J., Appleby, R. G., Abrahms, B., & Jordan, N. R. (2021). Animal-borne technologies in wildlife research and conservation. In S. A. Wich & A. K. Piel (Eds.), *Conservation technology* (pp. 105–127). Oxford University Press. https://doi.org/10.1093/oso/9780198850243.003.0006

How to cite this article: Lavender, E., Scheidegger, A., Albert, C., Biber, S. W., Illian, J., Thorburn, J., Smout, S., & Moor, H. (2025). patter: Particle algorithms for animal tracking in R and Julia. *Methods in Ecology and Evolution*, *00*, 1–8. https://doi.org/10.1111/2041-210X.70029