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

[2]School of Engineering Mathematics and Technology, University of Bristol

[3]School of Mathematics and Statistics, University of Glasgow

[4]School of Applied Sciences, Edinburgh Napier University

[5]Centre for Conservation and Restoration Science, Edinburgh Napier University

[6]Scottish Oceans Institute, University of St Andrews

[*]Correspondence: edward.lavender@eawag.ch

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, with substantive inputs from all authors. All authors contributed critically to the drafts and gave final approval for publication.

`patter` R package

Acknowledgments

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

Conflict of interest

The authors declare no conflicts of interest.

Data availability statement

The `patter` and `Patter.jl` packages are available on GitHub (Lavender, 2024a, 2024b). Manuscript code is archived on Zenodo (Lavender et al., 2024).

patter: particle algorithms for animal tracking in R and Julia

Abstract

1. In the field of movement ecology, state-space models have emerged as a powerful modelling framework that represents individual movements and the processes that connect movements to observations. However, fitting state-space models to animal tracking data is often 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 autonomous receiver arrays. patter is written in R, with a high-performance Julia backend. Package functionality supports data simulation, preparation, filtering, smoothing and mapping.
3. In two worked 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-based methods 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 two minutes during one month in minutes 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 home ranges, residency and habitat preferences.

`patter` `R` package

Keywords

`Julia`, movement ecology, passive acoustic telemetry, `patter`, `Patter.jl`, `R`

1. INTRODUCTION

The field of movement ecology has attracted huge interest in recent decades (Nathan et al., 2008; Rafiq et al., 2021). An ever-increasing suite of electronic tagging and tracking technologies is 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 autonomous receiver networks, known as passive acoustic telemetry arrays, have been established to track acoustically tagged benthic, demersal and pelagic taxa from local to continental scales (Matley et al., 2022). However, it is widely suggested that the accumulation of animal tracking data is outpacing the development of modelling methods and software packages for analysis (Rafiq et al., 2021).

In the last two decades, state-space models (SSMs) have emerged as a powerful modelling framework for animal tracking data (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 ($t \in \{1, \dots, T\}$) is imperfectly observed, generating 'noisy' observations ($\boldsymbol{y}_t$). In the context of animal tracking, SSMs model the movement process by which an animal's location ($\boldsymbol{s}$) evolves in (discrete) time (that is, $f(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1})$) and the observation processes that connect movements to observations (that is, $f(\boldsymbol{y}_t \mid \boldsymbol{s}_t)$). The SSM thus forms a formal statistical framework within which it is possible to estimate the unobserved locations of an animal that are of interest, whilst accounting for properties of movement (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 the context of animal tracking, a particle filter approximates the distribution of possible locations of an individual with a set of weighted samples termed 'particles' (Lavender et al., in prep). A movement model simulates possible locations of the individual (that is, $\boldsymbol{s}_t \sim f(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1})$) and observation model(s) weight particles in line with the probability of the observations (that is, $f(\boldsymbol{y}_t \mid \boldsymbol{s}_t)$). 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 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 ($f(\boldsymbol{s}_{1:T} \mid \boldsymbol{y}_{1:T})$), respectively (Doucet & Johansen, 2009). Compared to alternative SSM-fitting methods for animal-tracking data, advantages of particle-based methods 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 demersal fish geolocation at coarse spatial scales (over hundreds of kilometres) (Liu et al., 2019). However, existing routines are relatively specialised, computationally intensive and require significant user expertise.

Here, we introduce `patter`, an package that provides advanced particle filtering and smoothing algorithms for animal tracking data, motivated by our research in acoustic telemetry systems (Lavender, 2024a) (Lavender et al., in prep). The package is written in R (R Core Team, 2023) and integrates our high-performance Julia backend, Patter.jl (Lavender, 2024b). Julia is a relatively new 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. The package stands alongside generic SSM packages (King et al., 2016) but is designed for animal tracking—extending the R package ecosystem of this field (Joo et al., 2020) and that of the passive acoustic telemetry sub-field in particular (Kraft et al., 2023), with generic, flexible and fast particle algorithms. 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 residency, home ranges and habitat preferences.

2. METHODOLOGY

2.1. Model formulation

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

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

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 an unrestricted, 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 (turning angle) are independently distributed random variables.

Likelihood. The likelihood measures the probability of the observations given the latent locations ($\boldsymbol{s}_t$). In an acoustic telemetry system, observations include acoustic measurements at each of M receivers (an $M \times T$ matrix, $\boldsymbol{y}^{(A)}$), and/or ancillary data, such as depth measurements (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 $f(\boldsymbol{y}_t \mid \boldsymbol{s}_t) = f\left(\boldsymbol{y}_t^{(A)} \middle| \boldsymbol{s}_t\right) f\left(y_t^{(D)} \middle| \boldsymbol{s}_t\right)$.

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

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

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

where $z_t = \left(\varepsilon_{\text{deep}}(\boldsymbol{s}_t) + \ \varepsilon_{\text{shallow}}(\boldsymbol{s}_t\ )\right)^{-1}$. This requires $y_t^{(D)}$ to be within an envelope around the bathymetric depth $b(\boldsymbol{s}_i)$ defined by shallow- and deep depth-adjustment functions, $\varepsilon_{\text{shallow}}(\boldsymbol{s}_t), \varepsilon_{\text{deep}}(\boldsymbol{s}_t) \leq b(\boldsymbol{s}_t)$. These functions represent observational and spatially explicit bathymetric uncertainty and can be tailored for different species. For benthic species, small errors ($\varepsilon_{\text{shallow}}(\boldsymbol{s}_t), \varepsilon_{\text{deep}}(\boldsymbol{s}_t) \ll b(\boldsymbol{s}_t)$) only permit observations close to the seabed; for pelagic species larger $\varepsilon_{\text{shallow}}(\boldsymbol{s}_t)$ values permit observations in the water column.

2.2. Model inference

Filter. For inference, we begin with the partial marginal distribution, $f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:t})$. Particle filters approximate $f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:t})$ as a sum of N weighted particles, i.e., $f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:t}) \approx \sum_i^N \delta\left(\boldsymbol{s}_t - \boldsymbol{s}_{i,t}\right) w_i$, where δ is the Dirac delta function, w denotes normalised weights and i indexes particles. We assume static parameters are known. Starting with an initial set of particles sampled from the prior (i.e., $\boldsymbol{s}_{i,t=1} \sim f(\boldsymbol{s}_{t=1})$), the filter iteratively simulates particles via the movement model (i.e., $\boldsymbol{s}_{i,t} \sim f\left(\boldsymbol{s}_{i,t} \middle| \boldsymbol{s}_{i,t-1}\right)$), weights particles in line with the likelihood (via $w_{i,t} \propto w_{i,t-1} f(\boldsymbol{y}_t \mid \boldsymbol{s}_{i,t})$) and resamples particles accordingly.

Smoother. Particle smoothers re-weight filtered particles to approximate the full marginal, $f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:T})$. The two-filter smoother uses $\boldsymbol{s}_{i,\ldots,N,t}$ particles from a forward filter (with weights $w_{i,t}$) and $\tilde{\boldsymbol{s}}_{j,\ldots,N,t}$ particles from a backward filter (with weights $\widetilde{w}_{t,j}$). The distribution $f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:T})$ is approximated as a sum of re-weighted particles via $f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:T}) \approx \sum_j^N \delta(\boldsymbol{s}_t -$

$\tilde{\boldsymbol{s}}_{j,t}) \, \tilde{w}_{j,t|T}$, where the smoothed weights for each particle $\tilde{\boldsymbol{s}}_{j,t}$ represent a summation over all possible movements from preceding particles on the forward filter, that is, $\tilde{w}_{j,t|T} \approx \tilde{w}_{j,t} \sum_{i}^{N} f(\tilde{\boldsymbol{s}}_{j,t} \mid \boldsymbol{s}_{i,t-1}) \, w_{i,t-1}$. We use samples from $f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:T})$ to map utilisation distributions (Lavender et al., in prep). Sampling efficiently from the joint distribution, $f(\boldsymbol{s}_{1:T} \mid \boldsymbol{y}_{1:T})$, is more challenging and beyond the scope of this contribution.

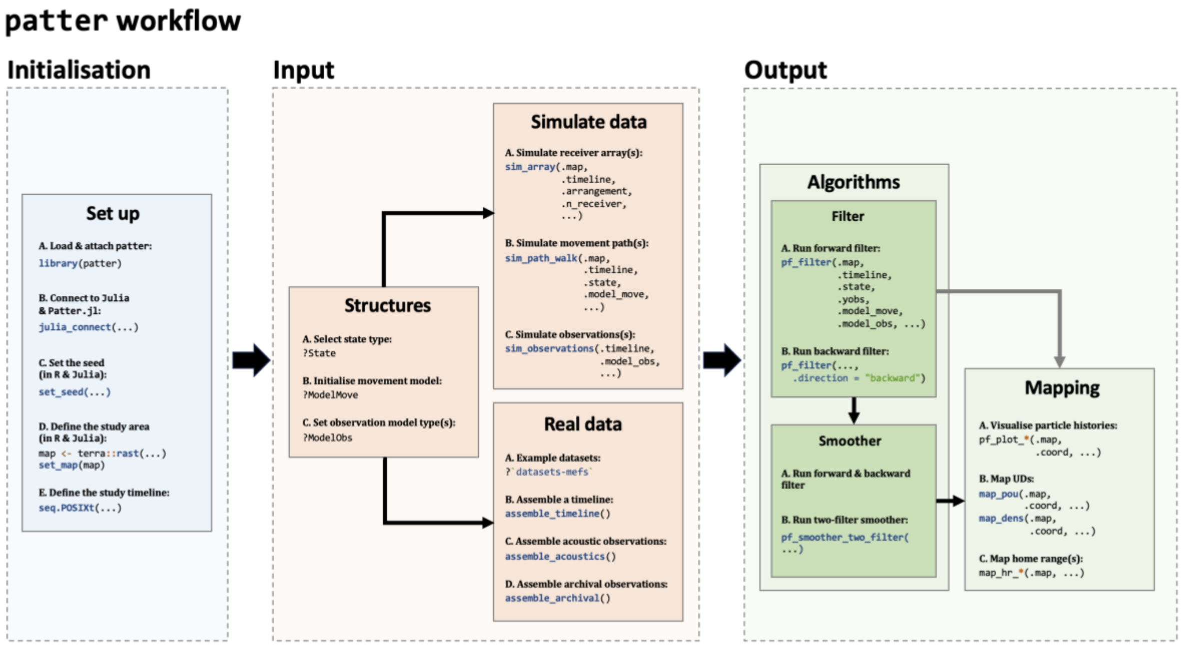
3. PACKAGE

`patter` supports data input, algorithms and mapping (Fig. 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 archival (depth) observations, but algorithm components can be customised as required. The main output of particle routines is a `data.table` of particles. The core algorithm routines wrap our high-performance `Patter.jl` package. `JuliaCall` implements the coupling between `patter` and `Patter.jl` (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.



A. Load & attach patter:

```
library(patter)
```

B. Connect to Julia & Patter.jl:

```
julia_connect(...)
```

C. Set the seed (in R & Julia):

```
set_seed(...)
```

D. Define the study area (in R & Julia):

```
map <- terra::rast(...)
set_map(map)
```

E. Define the study timeline:

```
seq.POSIXt(...)
```

A. Select state type:

```
?State
```

B. Initialise movement model:

```
?ModelMove
```

C. Set observation model type(s):

```
?ModelObs
```

A. Simulate receiver array(s):

```
sim_array(.map,
          .timeline,
          .arrangement,
          .n_receiver,
          ...)
```

B. Simulate movement path(s):

```
sim_path_walk(.map,
              .timeline,
              .state,
              .model_move,
              ...)
```

C. Simulate observations(s):

```
sim_observations(.timeline,
                 .model_obs,
                 ...)
```

A. Example datasets:

```
?`datasets-mefs`
```

B. Assemble a timeline:

```
assemble_timeline()
```

C. Assemble acoustic observations:

```
assemble_acoustics()
```

D. Assemble archival observations:

```
assemble_archival()
```

A. Run forward filter:

```
pf_filter(.map,
          .timeline,
          .state,
          .yobs,
          .model_move,
          .model_obs, ...)
```

B. Run backward filter:

```
pf_filter(...,
  .direction = "backward")
```

A. Run forward & backward filter

B. Run two-filter smoother:

```
pf_smoother_two_filter(
...)
```

A. Visualise particle histories:

```
pf_plot_*(.map,
          .coord, ...)
```

B. Map UDs:

```
map_pou(.map,
        .coord, ...)
map_dens(.map,
         .coord, ...)
```

C. Map home range(s):

```
map_hr_*(.map, ...)
```

Figure 1. Overview of the `patter` package.

4. EXAMPLES

4.1. Overview

We provide two worked examples using simulated data. Simulation is a useful starting point for illustration and real-world analyses, where it informs algorithm implementation and interpretation. In both examples, we consider the movements of a benthic animal in a hypothetical acoustic array spanning a Marine Protected Area (MPA) in Scotland (Fig. 2). The study area is defined by a 100 x 100 m 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 two minutes and simulate a discrete-time random walk at this resolution over a one-month period (Fig. 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, we simulate observations in such a way that the depth observation exactly defines the location of the individual (which is situated on the seafloor) (Fig. 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-based methods and the reconstruction of maps of space use (Fig. 2D–F). In the following sections, we showcase key functions and arguments (Fig. 1). Additional arguments are denoted by ellipses. Complete code is available online (Lavender et al., 2024).

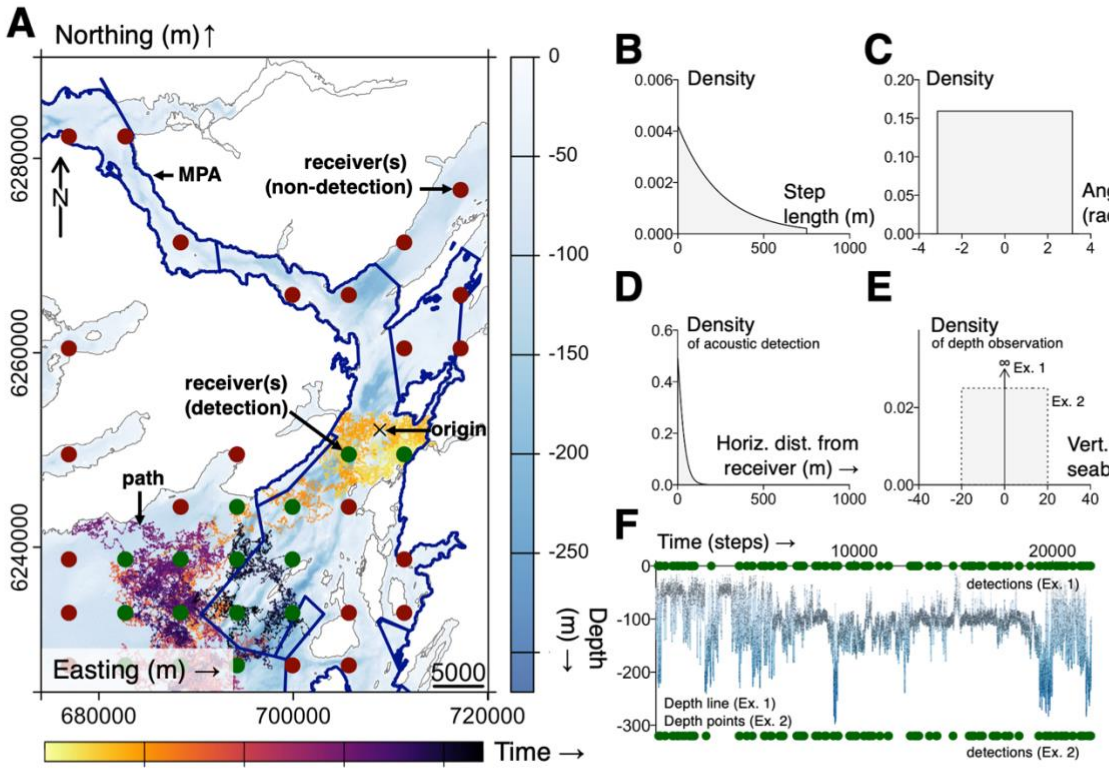


Figure 2. The components of a state-space model for animal tracking data. A shows the study area, including a simulated movement path (coloured by time) and acoustic receivers (sized by detection range and coloured by detection(s)/non-detection). **B–C** show the components of the random walk used to simulate and model movements. **D–E** show 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 §4.1). **F** shows the simulated time series for each example.

4.2. Implementation

4.2.1. Simulation

We begin with essential initiation:

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

Next, we define the study system:

```r
# 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:

```r
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}), subject to boundary conditions)
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 archival 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 $\varepsilon_{\text{shallow}}(\boldsymbol{s}_t)$ and $\varepsilon_{\text{deep}}(\boldsymbol{s}_t)$ 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:

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

In the second example, we incorporate uncertainty:

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

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

```r
model_obs <- c("ModelObsAcousticLogisTrunc","ModelObsDepthUniform")
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", ...
)
```

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(...)
```

`patter` R package

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. Users can define species-specific movement models (of any dimension), include diverse observation types and implement system-specific observation models. See the package documentation for details.

4.3. Results

In our first example, the particle filter reconstructs the simulated movement path perfectly (Fig. 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 the partial marginal distribution, $f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:t})$ (Fig. 3B). The particle smoother re-weights filtered particles, approximating the full marginal, $f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:T})$ (Fig. 3C). Smoothed particles can be used to map patterns of space use, estimate home ranges and quantify residency (Fig. 3D). The quality of the smoothing depends on the filter. In this case, filter diagnostics are adequate (Fig. 3E). Total computation time ranged from 5–32 minutes for examples 1–2 on a 2023 MacBook Pro (Apple M2 Pro, 32 GB RAM, 12 CPUs).

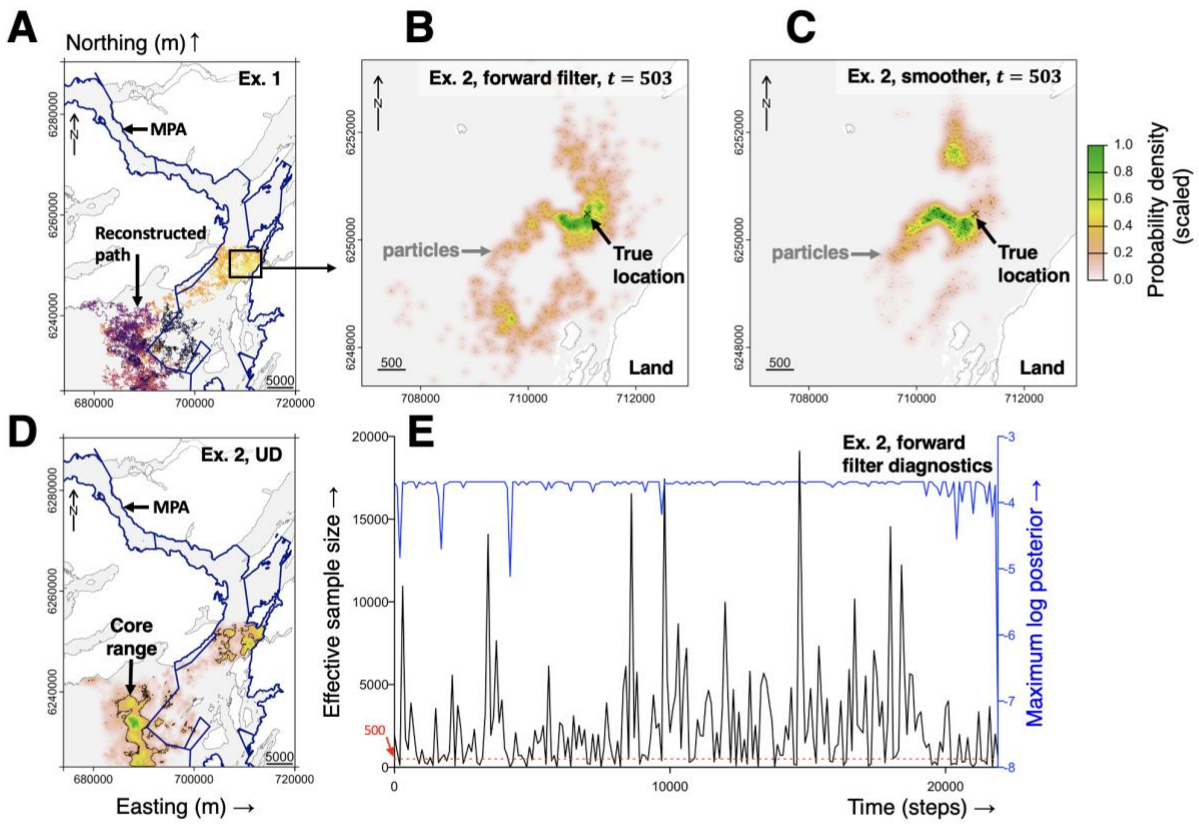


Figure 3. `patter` outputs for the first (A) and second (B–E) worked examples. A is the movement path reconstructed by the particle filter for the first example. The simulated path is recovered perfectly (at grid resolution) because the observations exactly define the individual's location. **B**–**C** 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.

5. DISCUSSION

`patter` R package

`patter` provides a robust, fast and accessible implementation of particle-based methods for animal tracking data (Lavender et al. in prep). In the context of passive acoustic telemetry, `patter` is unique in the provision of algorithms that represent the movement and observation processes that generate observations, including movement capacity, barriers to movement, detections and ancillary observations, within a biologically and statistically sound framework. The algorithms outperform standard heuristic analytical approaches (provided by packages such as `VTrack` and `RSP`) (Lavender et al., in prep). Furthermore, the movement and observation models are customisable, which makes 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., in prep).

A key feature of `patter` is the speed provided by the `Patter.jl` backend (Lavender, 2024b). Previous research has linked the underutilisation of particle algorithms in ecology to their computational requirements (Liu et al., 2019), which are typically $O(NT)$ for filtering and $O(N^2T)$ for smoothing (Doucet & Johansen, 2009). While adequate filtering is often possible with relatively few particles ($N \approx 1000$), this imposes some practical constraints that can be limiting in situations where numerous particles are required to ensure convergence (namely, labyrinthine landscapes where a tiny fraction of possible routes are compatible with the data) (Lavender et al., in prep). The particle penalty is more severe for smoothing but mitigated via subsampling, since in general only a subset of the particles required for a successful filter run are ultimately required to approximate the distribution of latent locations (in two or three dimensions). In practice, our packages achieve speeds that compare favourably with other geolocation routines (even with ≤ 1 million particles). For example, Hostetter & Royle (2020) formulated a state-space model for acoustic detections ($T = 150$) alongside a bespoke Jags implementation that requires ~15 hours on a standard computer to run. With our particle filtering–smoothing algorithm, the estimation of latent locations in this situation is soluble in seconds. Other geolocation packages developed for demersal fish that fit hidden Markov models using maximum likelihood typically require hours or days to derive daily geolocation estimates for a one-year period and even a novel Python particle filter that exploits GPU-parallelisation takes up to an hour (Liu et al., 2019). 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 (Lavender et al., in prep).

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., in prep). Joint estimation of model parameters and latent locations is a possible future development. As an example real-world analysis, we are currently analysing acoustic and archival data from the Critically Endangered flapper skate (*Dipturus intermedius*) to quantify patterns of space use and site affinity to 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.

patter R package

References

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 *In 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), 1255642–10. 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

King, A. A., Nguyen, D., & Ionides, E. L. (2016). Statistical Inference for Partially Observed Markov Processes via the R Package pomp. *Journal of Statistical Software*, *69*(12), 1–43. https://doi.org/10.18637/jss.v069.i12

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* (1.0.0) [R]. https://github.com/edwardlavender/patter

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

Lavender, E., Scheidegger, A., Albert, C., Biber, S., Illian, J., Thorburn, J., Smout, S., & Moor, H. (2024). *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*, *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

R Core Team. (2023). *R: a language and environment for statistical computing* (4.3.1) [Computer software]. R Foundation for Statistical Computing. https://r-project.org

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