State-space modelling

in passive acoustic telemetry systems with `patter`

Edward Lavender
edward.lavender@eawag.ch

eawag
aquatic research

Acknowledgements

Ed Lavender

Helen Moor

Andreas Scheidegger

Carlo Albert

Eawag team

Stanisław Biber

Janine Illian

Sophie Smout

James Thorburn

Collaborators

Acknowledgements

eawag aquatic research

University of St Andrews

CREEM
Centre for Research into Ecological and Environmental Modelling

Scottish Oceans Institute

marine scotland science

University of Bristol

University of Surrey

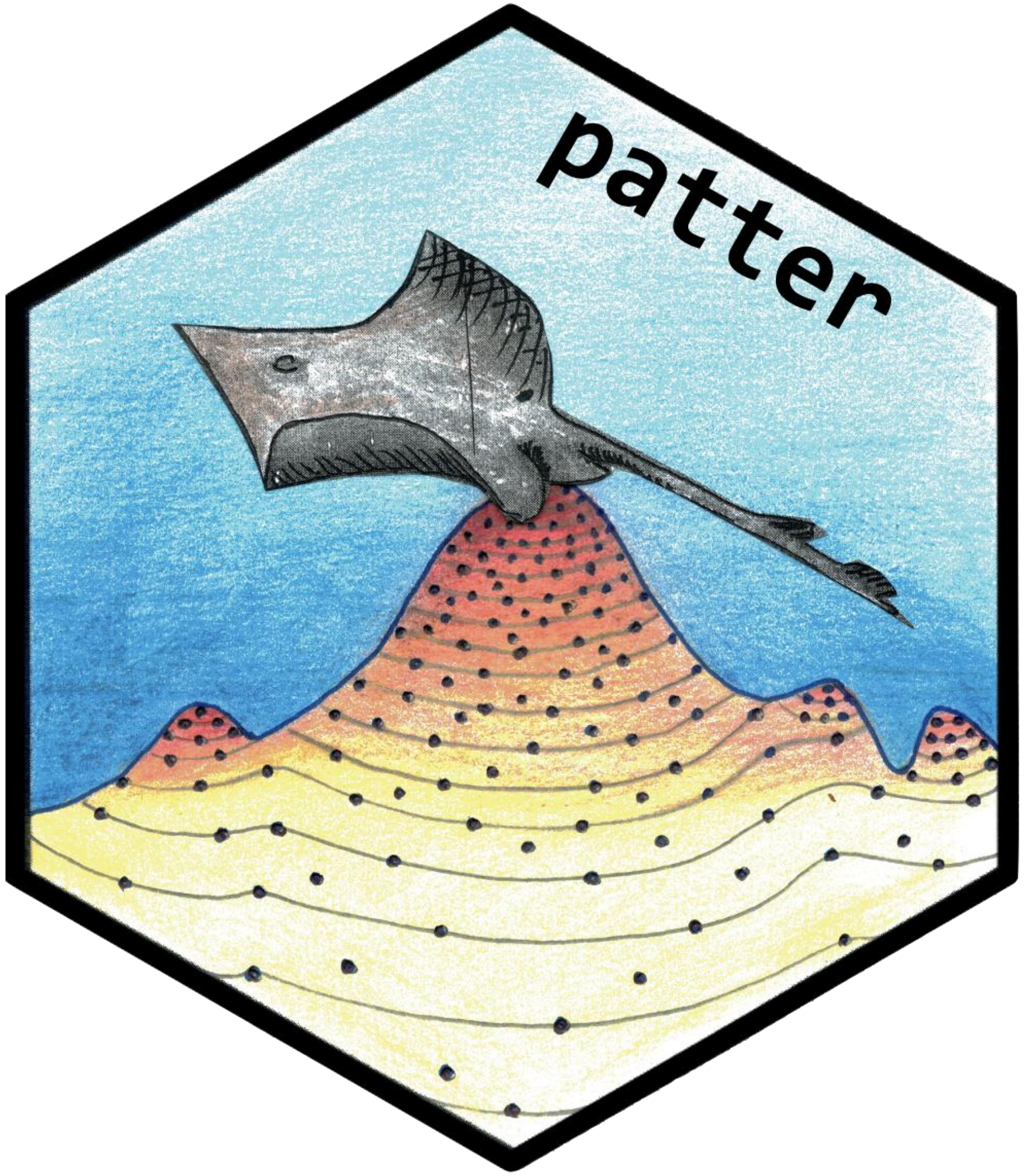
Shark Guardian

Marine Alliance Science Technology Scotland

NatureScot
Scotland's Nature Agency
Buidheann Nàdair na h-Alba

University of Glasgow
Via Veritas Vita

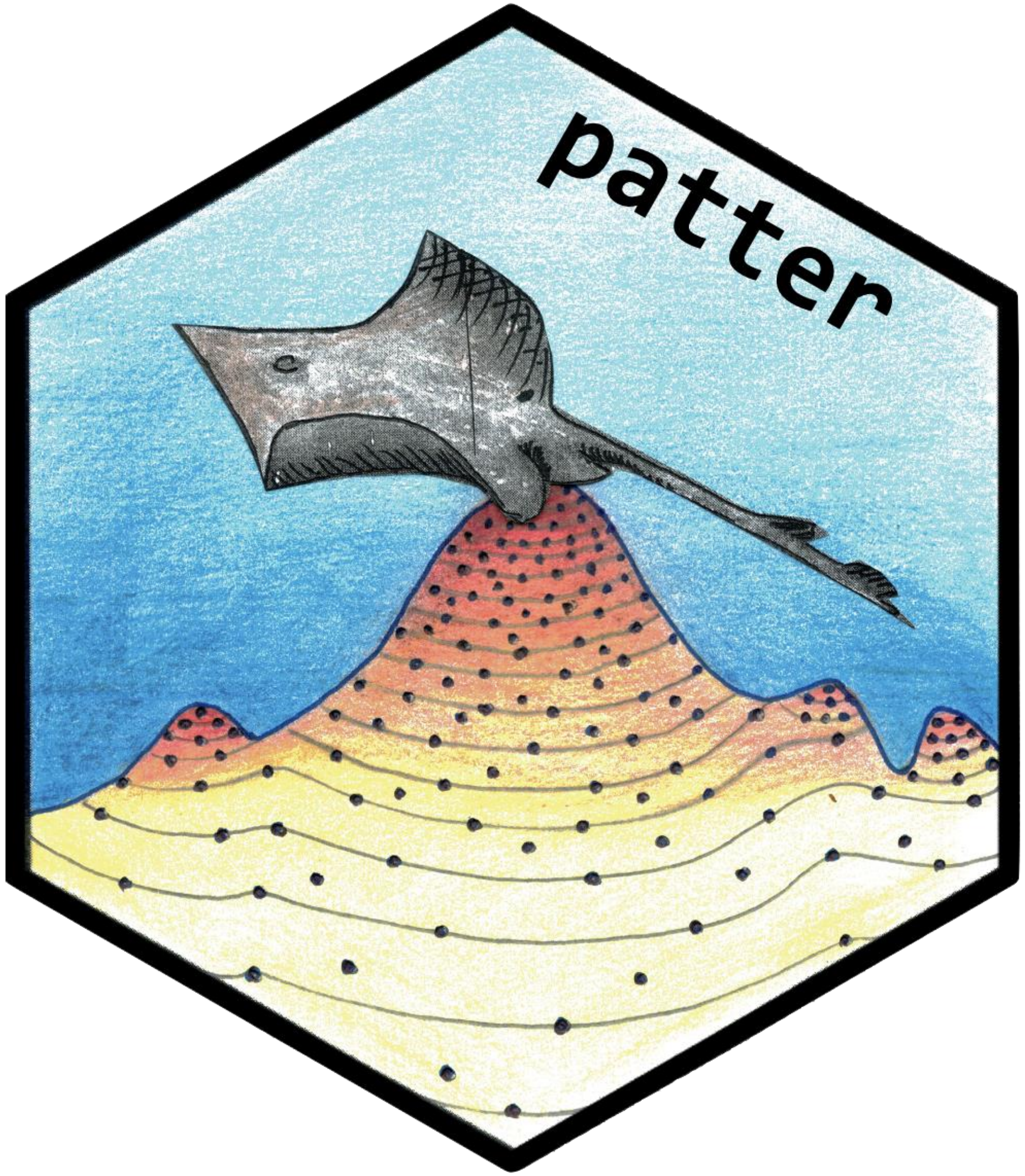
Contents



Take home messages

Take home messages

- `patter` is an R package that fits **state-space models** to animal-tracking data using **particle algorithms**
- State-space models bring together analyses of **movements**, patterns of **space** use, **residency** etc.
- By fitting models to data, we **estimate the states** (locations) of an individual through time and uncertainty
- Robust estimates of individual locations are a foundation for **downstream ecological analyses**

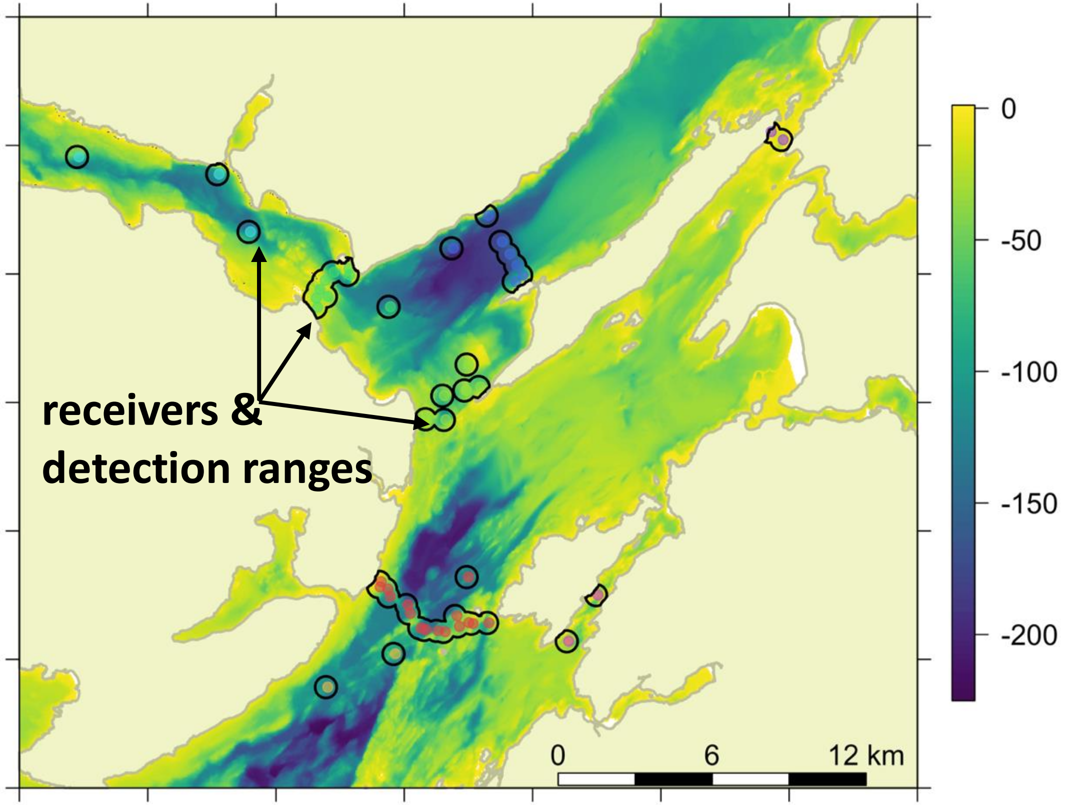


Introduction

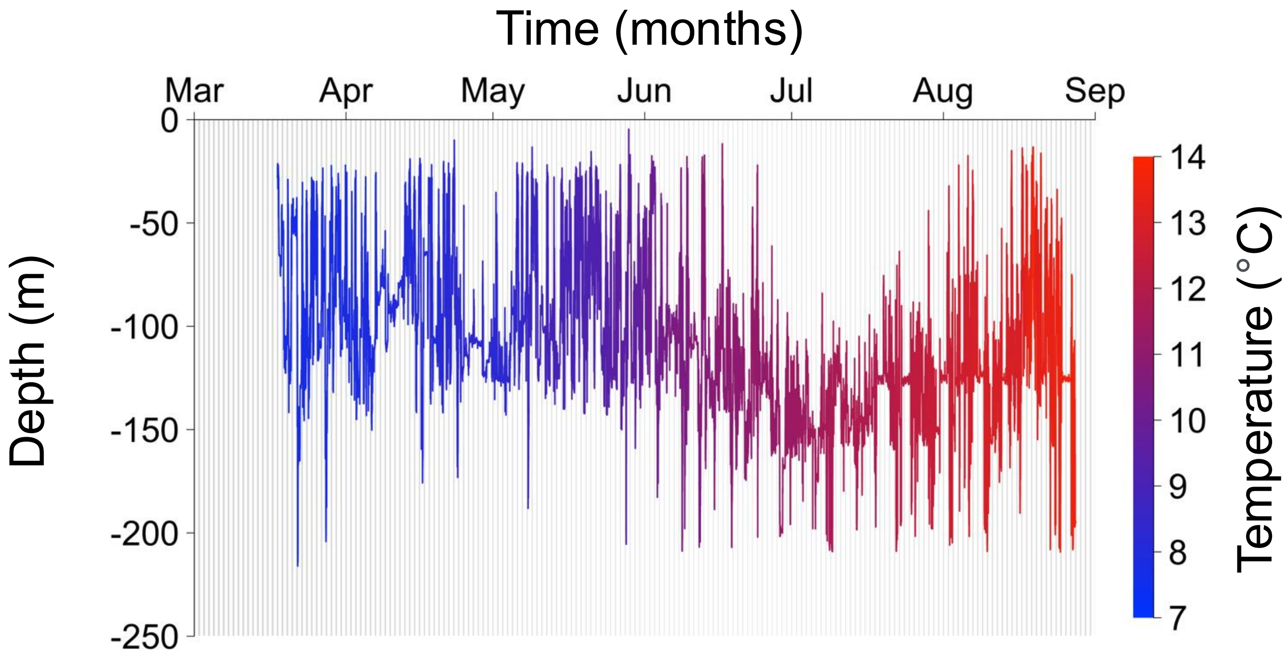
Introduction

This presentation focuses on passive acoustic telemetry & ancillary biologging sensors, but the methods are more widely applicable

Passive acoustic telemetry

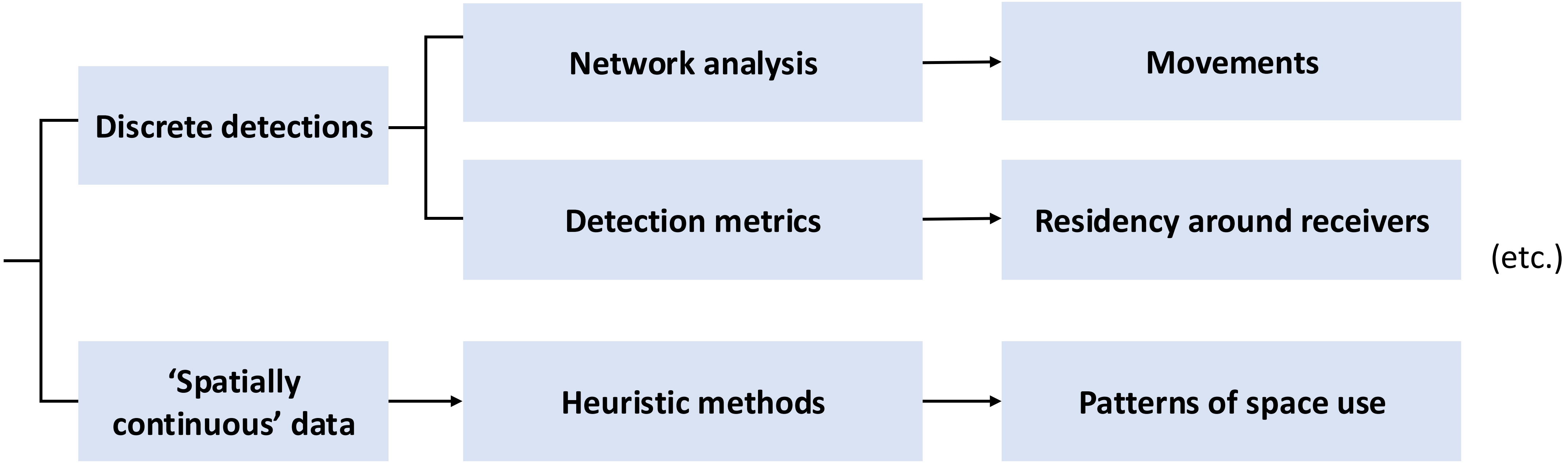


Archival time series



Introduction

Two broad approaches have emerged for the analysis of passive acoustic telemetry data

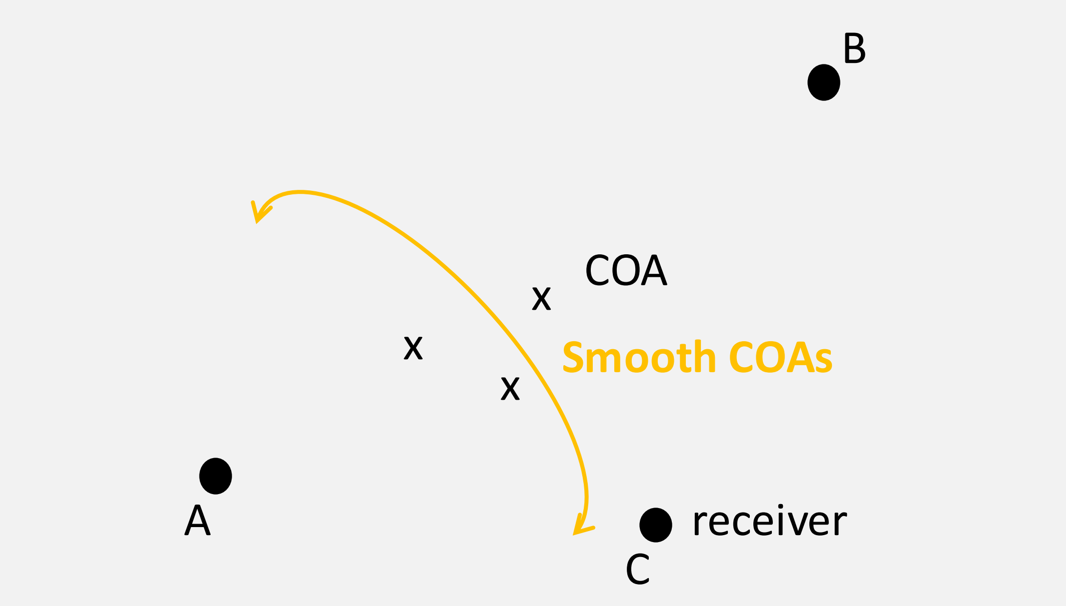


Whoriskey et al. (2019). Meth. Ecol. Evol.

Introduction

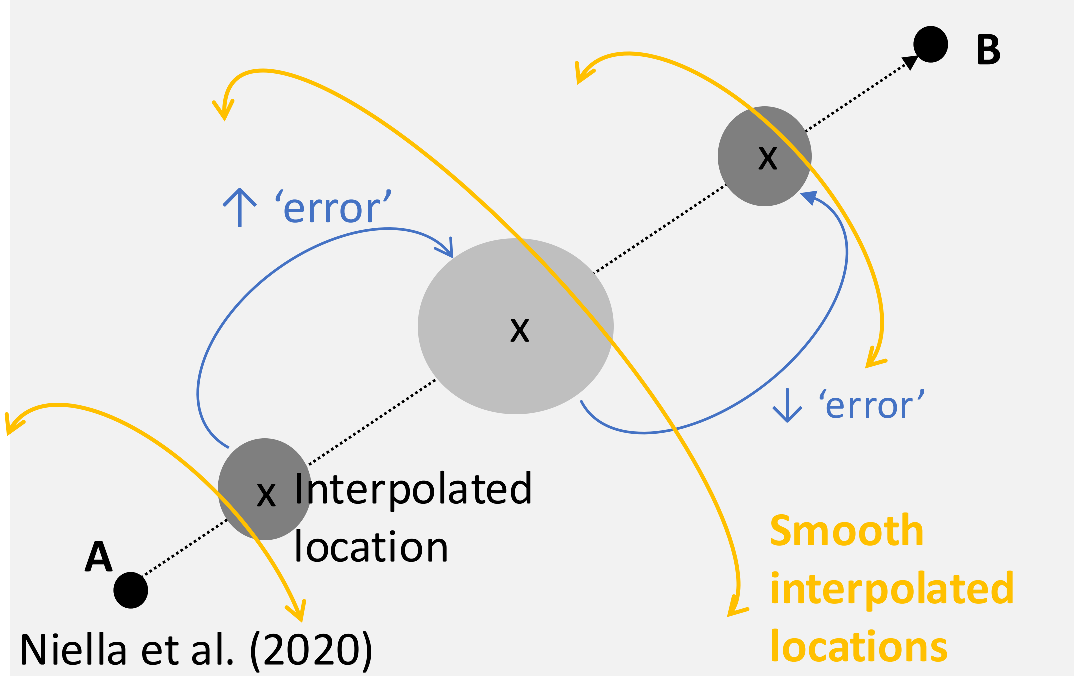
Heuristic methods are typically used for mapping space use

Mean positions



Simpfendorfer et al. (2002)

Refined shortest paths



Niella et al. (2020)

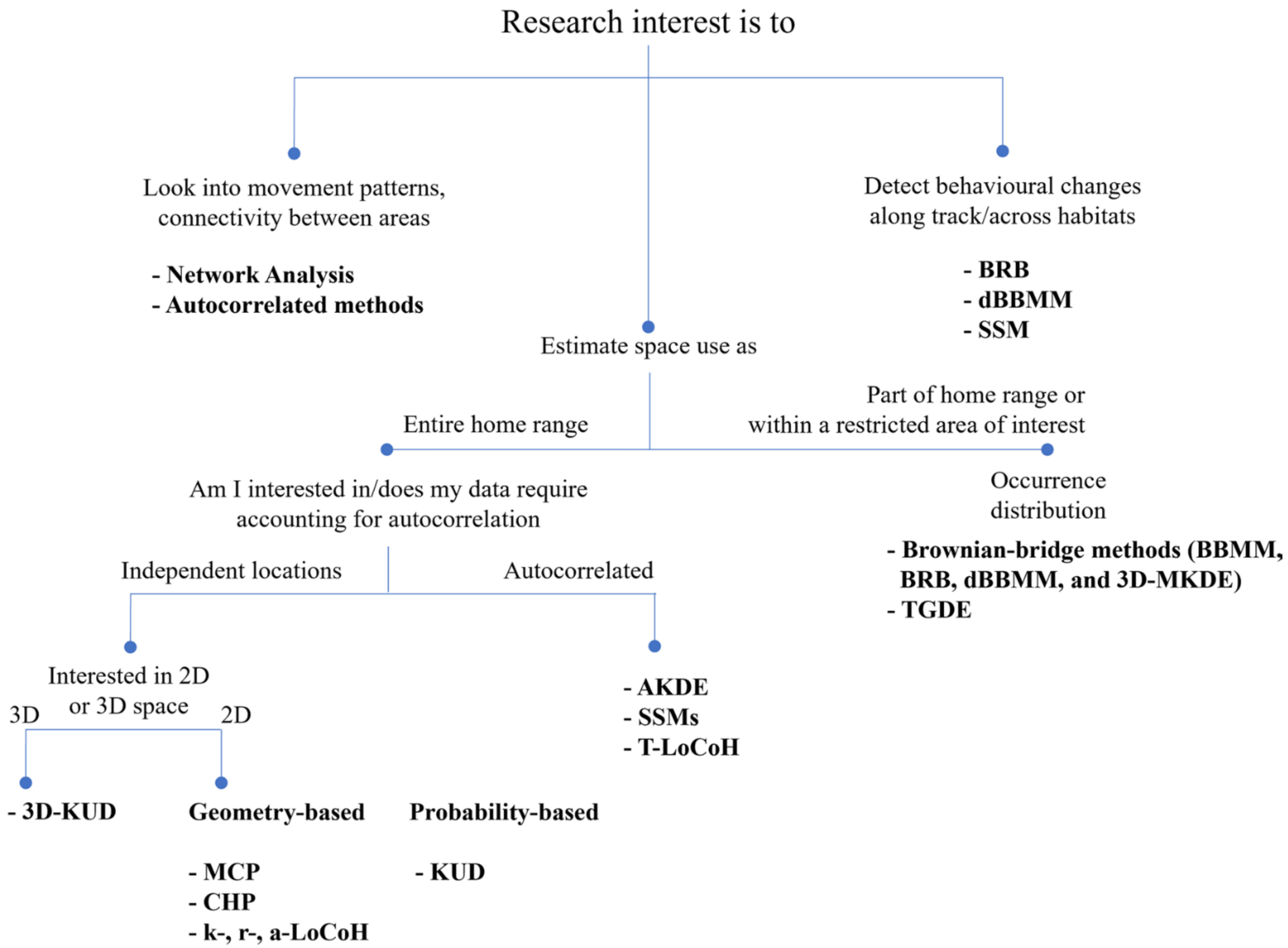
Introduction

Heuristic methods have pros & cons

Pros	**Cons**
Ease of implementation	Biological knowledge
Speed of implementation	Ancillary datasets
Descriptive analyses w/o uncertainty	Variable performance
	Interpretation
	Uncertainty quantification

Introduction

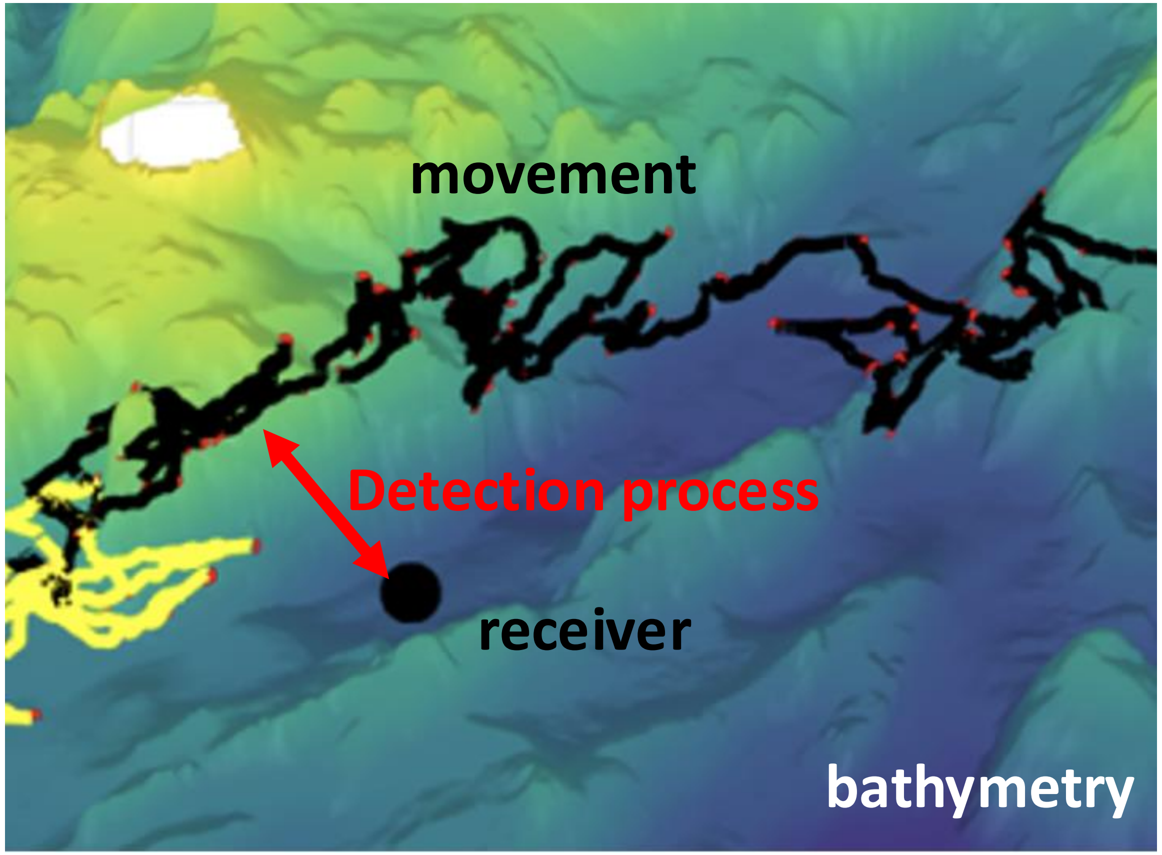
- Recent reviews structure guidance in decision trees or tables
- **Decision trees** link research questions to existing methodologies, but **draw distinctions** between analyses of **movement patterns** (e.g., network analysis), **space use** (e.g., kernel smoothing) **& residency** (e.g., residency metrics).



Kraft et al. (2023). Movement Ecology.

Introduction

We can bridge the divide between movement, space-use & residency analyses



Introduction

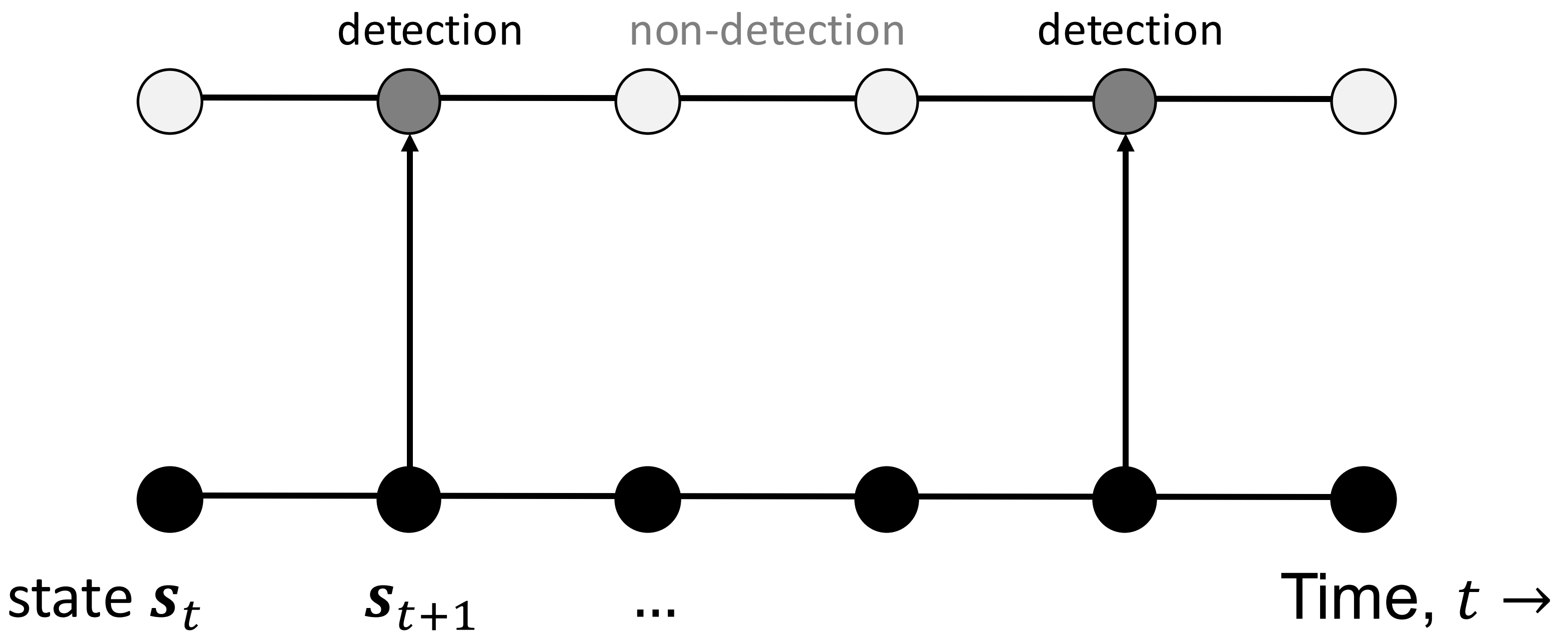
We can do this using state-space models

Observation process(es)

$\boldsymbol{y}_t \sim f(\boldsymbol{y}_t \mid \boldsymbol{s}_t)$

Movement process

$\boldsymbol{s}_t \sim f(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1})$

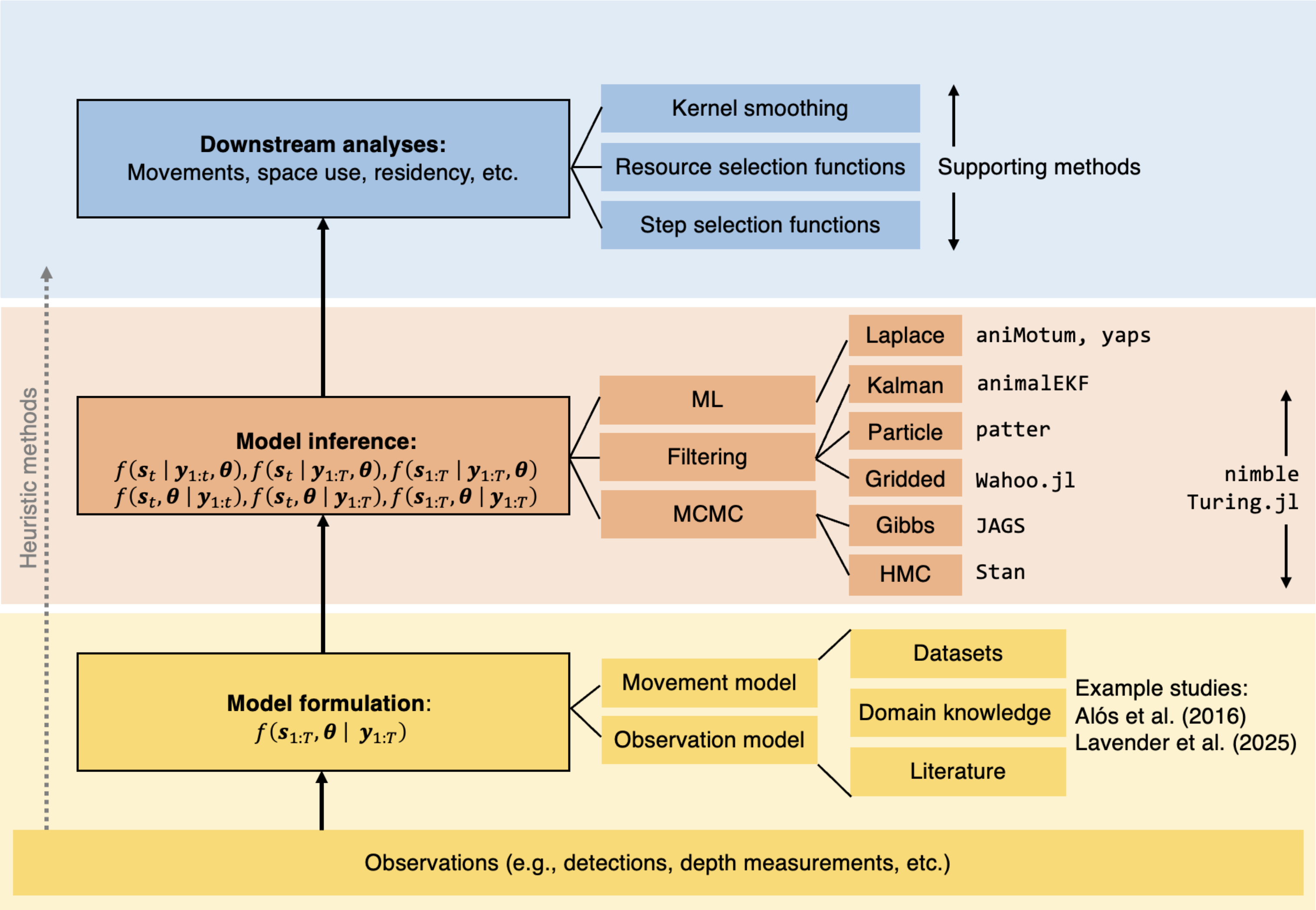


Introduction

- **State-space models** provide a biologically & statistically **solid foundation** for movement modelling in animal tracking studies

- By fitting a model to data, we can **estimate** the latent **states** & static **parameters**

- **Probabilistically sound estimates** of individual location are essential for **downstream analyses**



Lavender et al. (in review)

Introduction

There are many benefits of a model-based data analysis approach

- We can build analyses that link movements, space use & residency

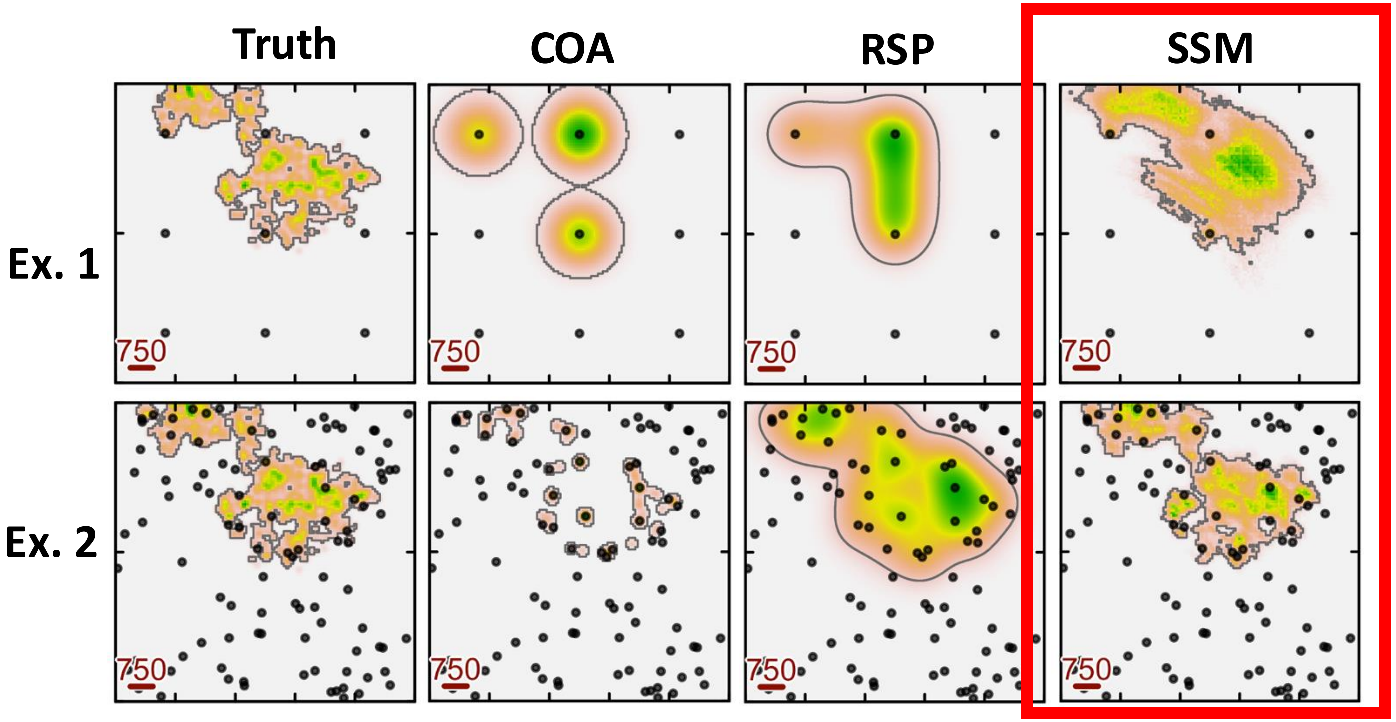
- We can leverage our **biological knowledge** & **disparate data types**

- We generate **probabilistic** estimates of latent states with **uncertainty quantification**

- **Sound estimates of individual location** are crucial for downstream analyses:
  - Movement patterns
  - Co-occurrence patterns
  - Maps of space use
  - Residency
  - Habitat selection

Introduction

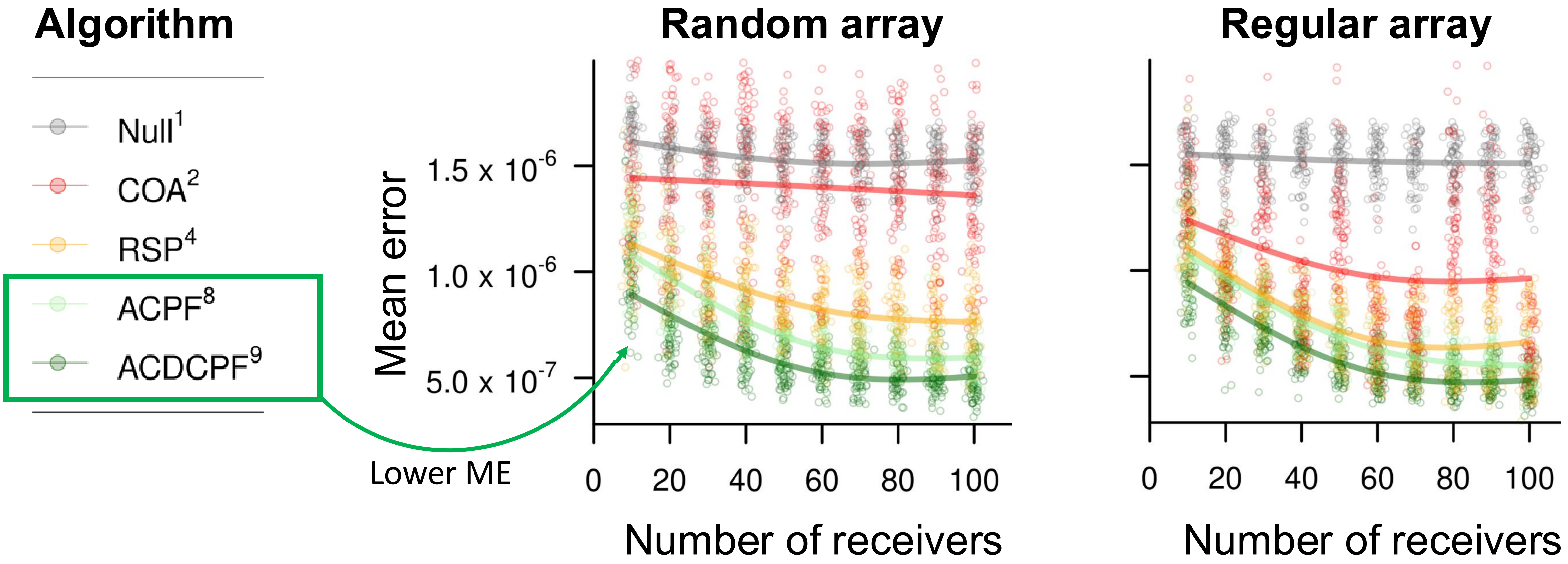
With a model-based approach, we generate improved maps of space use



Lavender et al. (2025). Meth. Ecol. Evol.

Introduction

Model-based inference algorithms outperform heuristic methods across the board

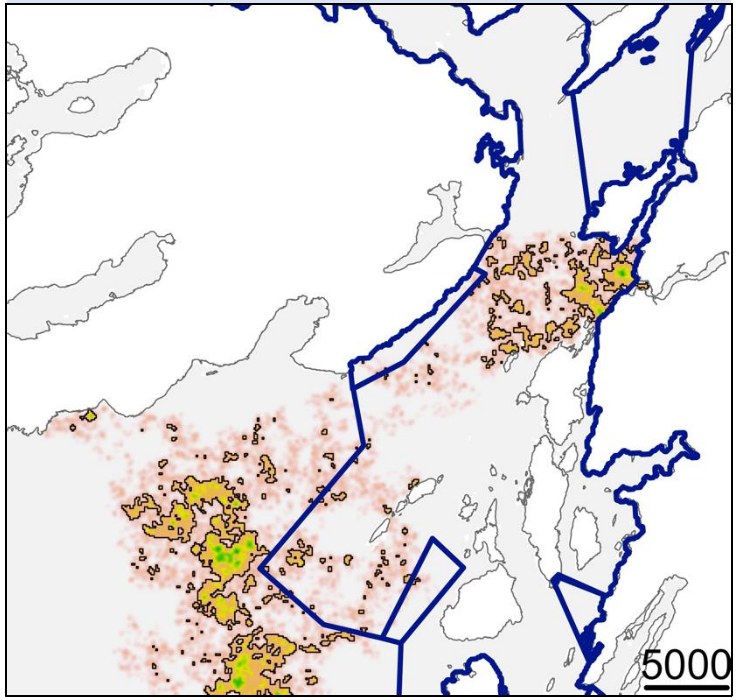


Lavender et al. (2025). Meth. Ecol. Evol.

Introduction

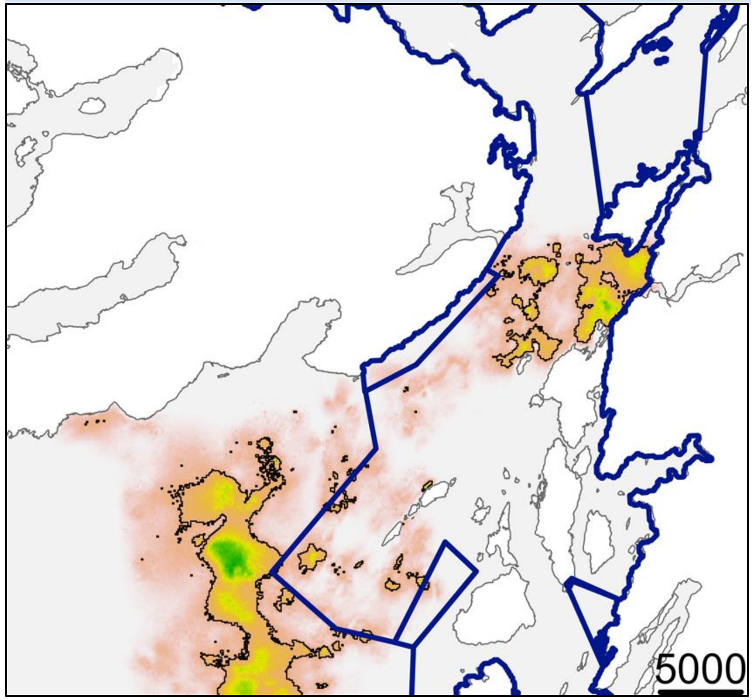
We can generate refined estimates of residency at daily, weekly & yearly timescales

‘True’ map



True **residency** in MPA: **37.0 %**

Reconstructed map

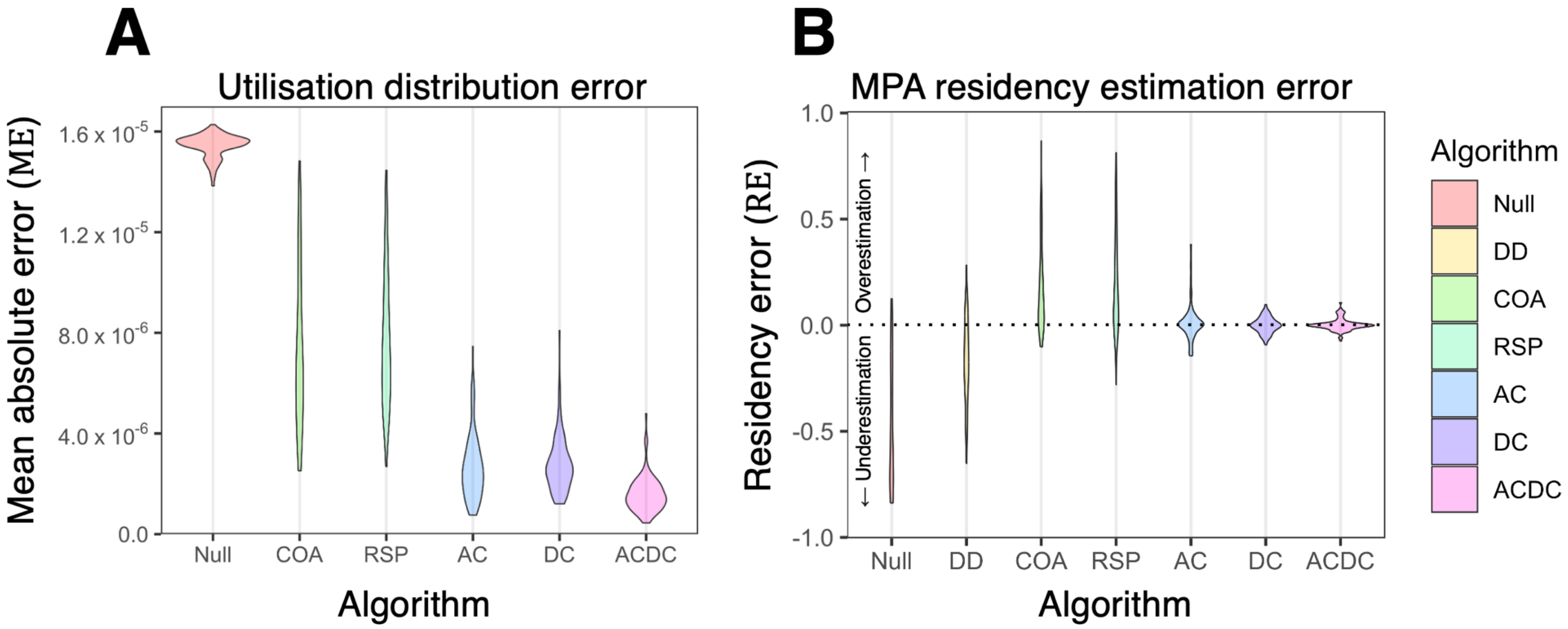


Estimated **residency**: **36.3 %**

Lavender et al. (2025). Meth. Ecol. Evol.

Introduction

In a case-study analysis for flapper skate, we found 5-fold improvements in maps of space use & >30-fold improvements in residency compared to heuristic methods

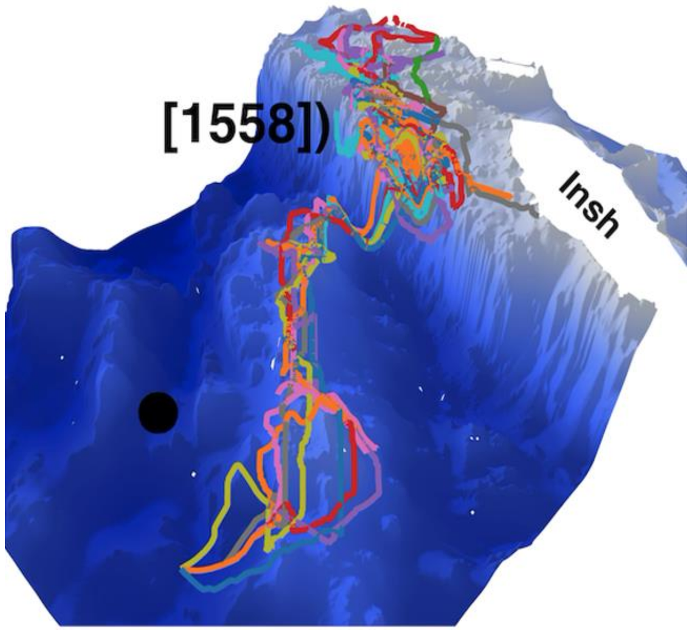


Lavender et al. (in review)

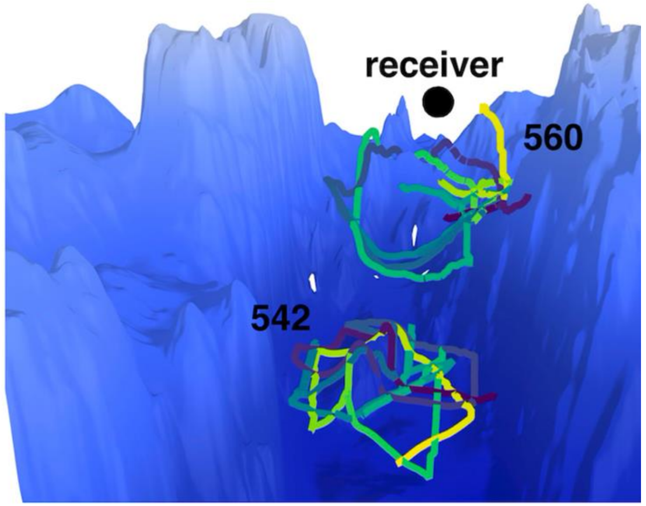
Introduction

Probabilistically sound estimates of individual location support refined downstream analyses

Individual movements



Co-occurrence



Lavender et al. (2023). Meth. Ecol. Evol.

Habitat selection

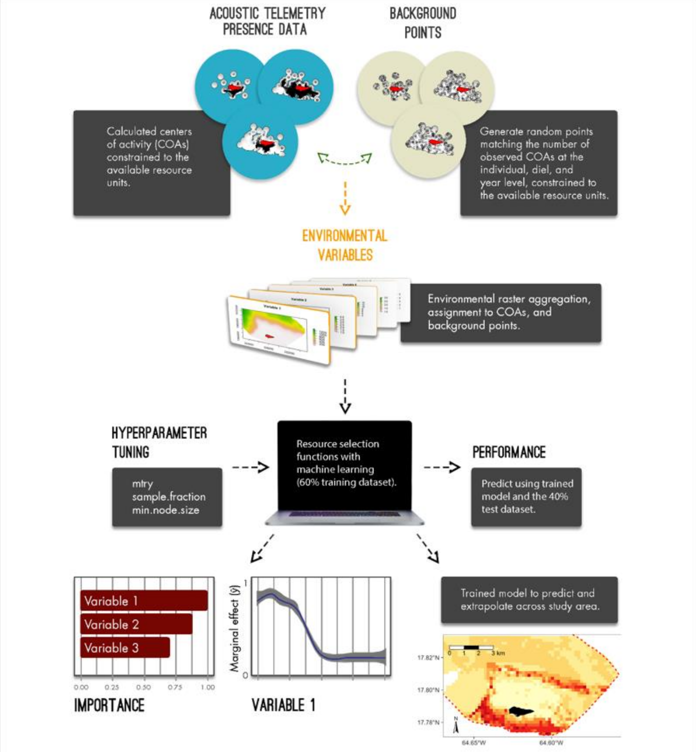


FIGURE 1 | Conceptual diagram to implement resource selection functions with acoustic telemetry and machine learning algorithms; from generated centers of activities and background points, deriving available resource units, model development, model performance and interpretation, and, ultimately, predicting and extrapolating relative selection.

Griffin et al. (2021). Front. Mar. Sci.

State-space model

State-space model

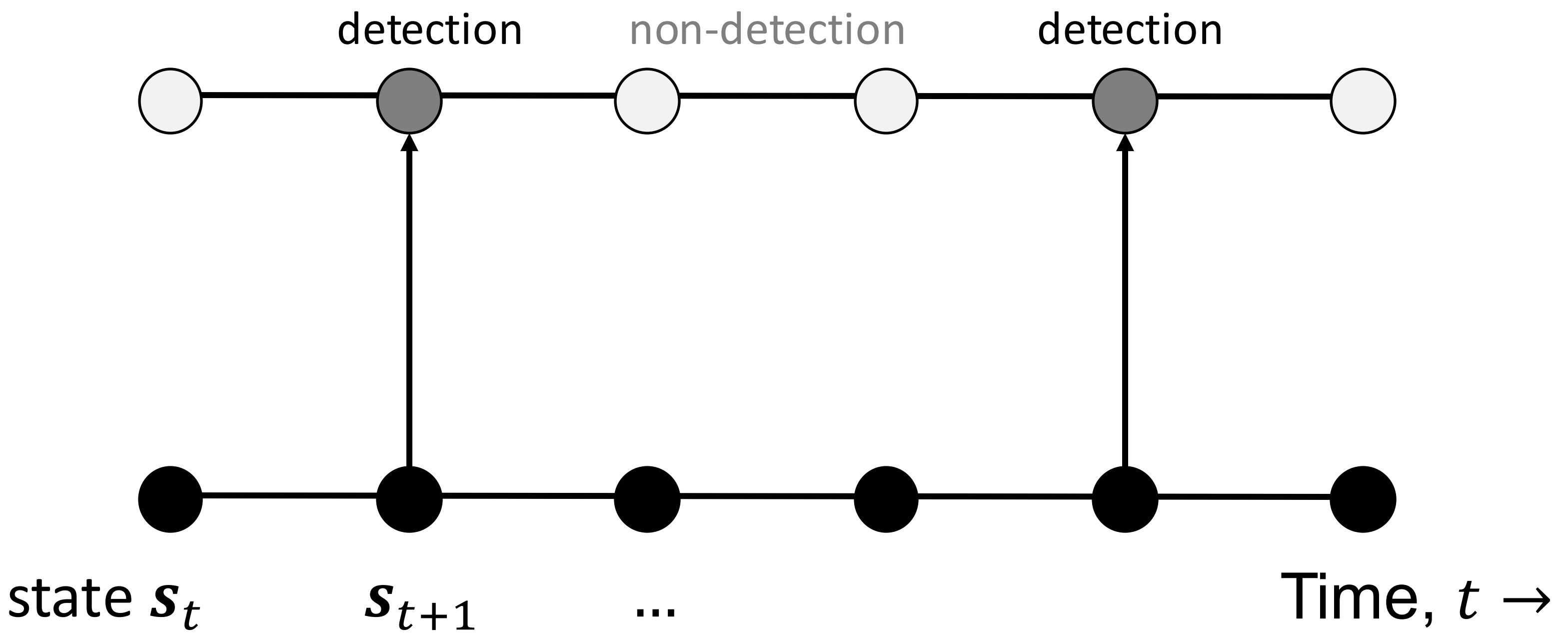
We can formulate a SSM that represents the movement & observation processes

Observation process(es)

$\boldsymbol{y}_t \sim f(\boldsymbol{y}_t \mid \boldsymbol{s}_t)$

Movement process

$\boldsymbol{s}_t \sim f(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1})$



State-space model

We can formulate a SSM that represents the movement & observation processes

$$f(\boldsymbol{s}_{1:T}, \boldsymbol{\theta} \mid \boldsymbol{y}_{1:T}) \propto f(\boldsymbol{s}_{1:T} \mid \boldsymbol{\theta})\, f(\boldsymbol{y}_{1:T} \mid \boldsymbol{s}_{1:T}, \boldsymbol{\theta})\, f(\boldsymbol{\theta})$$

Movement process

$$f(\boldsymbol{s}_{1:T} \mid \boldsymbol{\theta}) = f(\boldsymbol{s}_{t=1} \mid \boldsymbol{\theta}) \prod_{t=2}^{T} f(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1}, \boldsymbol{\theta})$$

$$f(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1}, \sigma) = N(\boldsymbol{s}_t;\ \boldsymbol{s}_{t-1}, \sigma^2 \boldsymbol{I})$$

Observation process

$$f(\boldsymbol{y}_{1:T} \mid \boldsymbol{s}_{1:T}, \boldsymbol{\theta}) = \prod_{t=1}^{T} f(\boldsymbol{y}_t \mid \boldsymbol{s}_t, \boldsymbol{\theta})$$

$$f\left(\boldsymbol{y}_t^{(A)} \mid \boldsymbol{s}_t, \boldsymbol{\theta}\right) = \prod_k f\left(y_{t,k}^{(A)} \mid \boldsymbol{s}_t, \boldsymbol{\theta}\right)$$

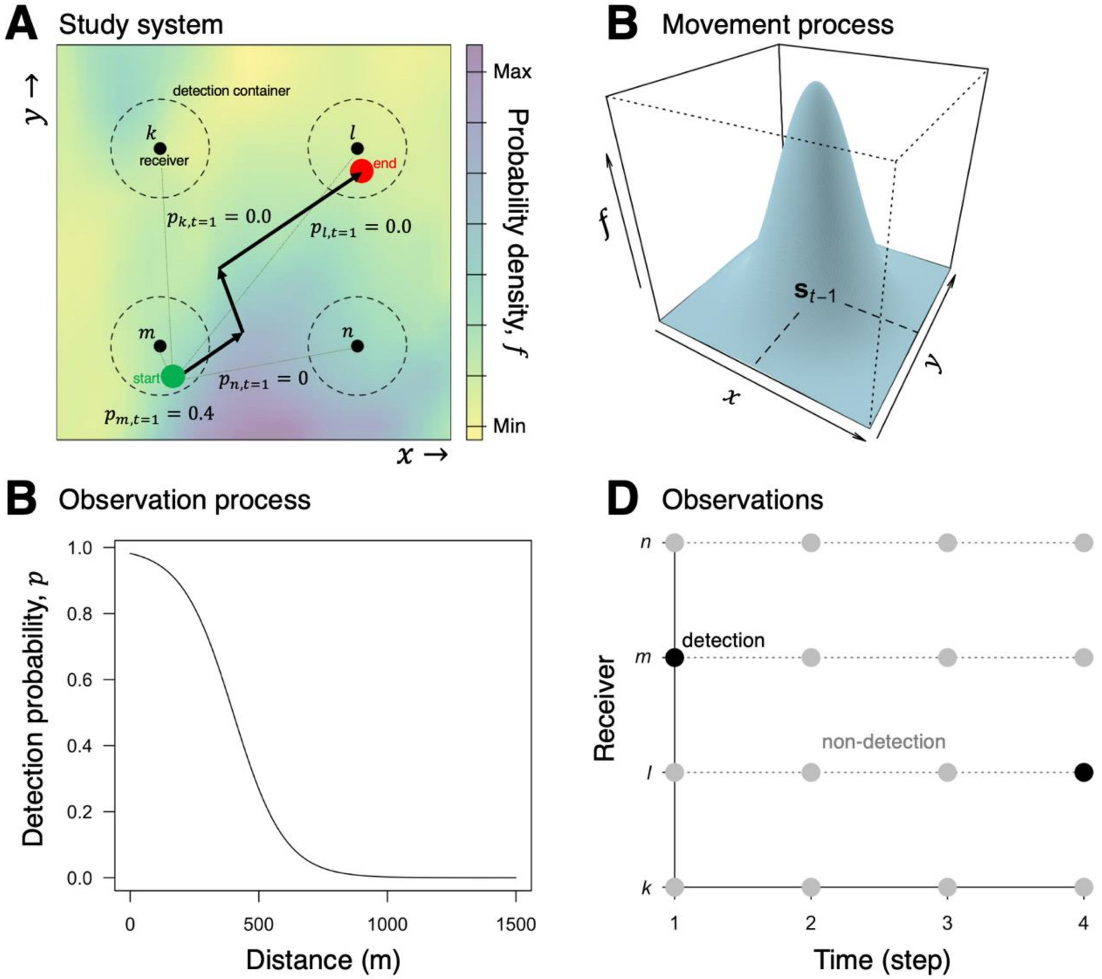
$$f\left(y_{t,k}^{(A)} \mid \boldsymbol{s}_t, \boldsymbol{\theta}\right) = \mathrm{Bernoulli}\left(p_{t,k}(\boldsymbol{s}_t, \boldsymbol{\theta})\right)$$

$$p_{t,k}(\boldsymbol{s}_t, \boldsymbol{\theta}) = \mathrm{logistic}(\mathrm{distance}(\boldsymbol{s}_t, \boldsymbol{r}_k), \boldsymbol{\theta})$$

Lavender et al. (in review)

State-space model

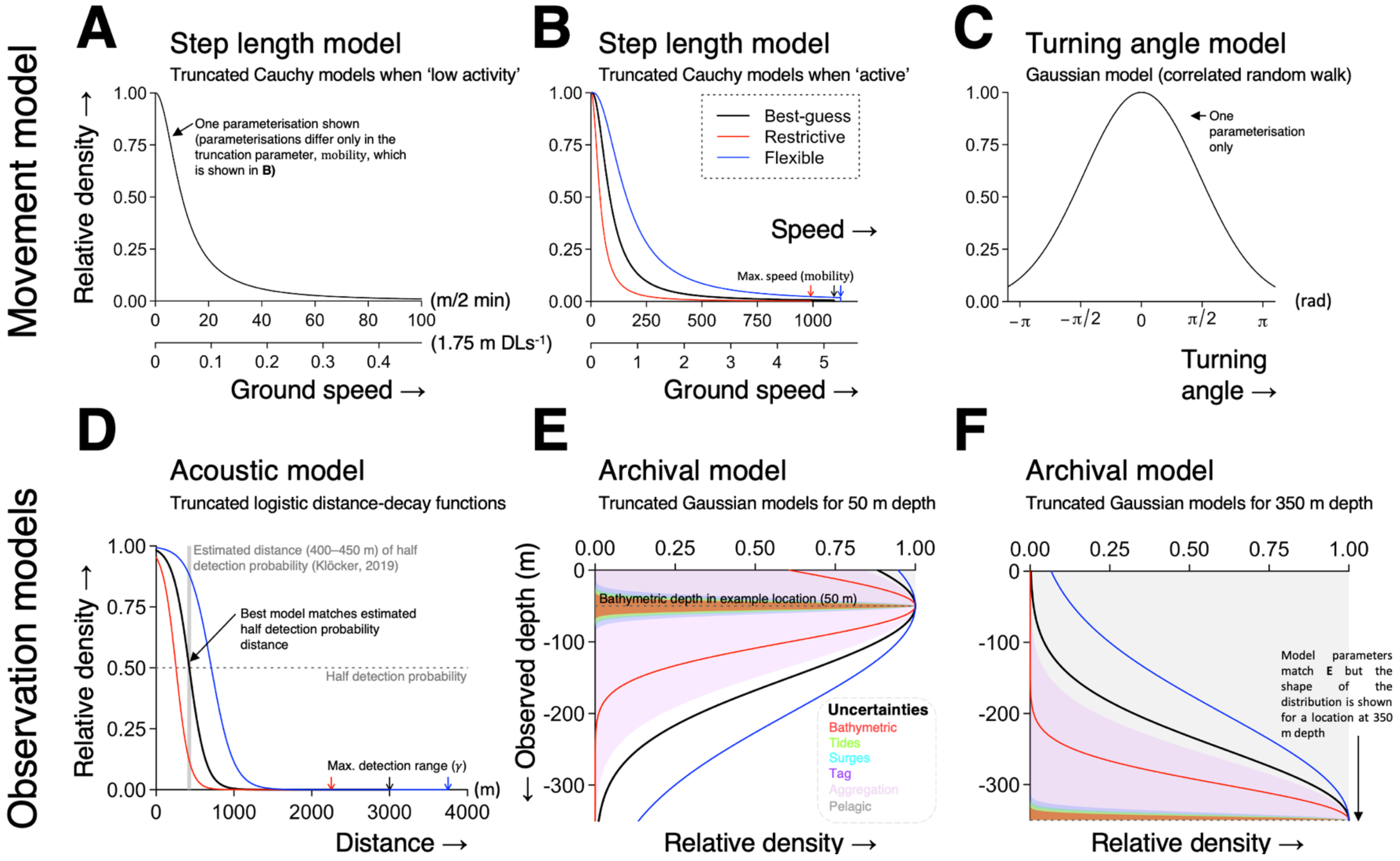
We can formulate a SSM that represents the movement & observation processes



Lavender et al. (in review)

State-space model

State-space models should leverage biological expertise, available information & literature



Lavender et al. (in review)

Model-based inference

Model-based inference

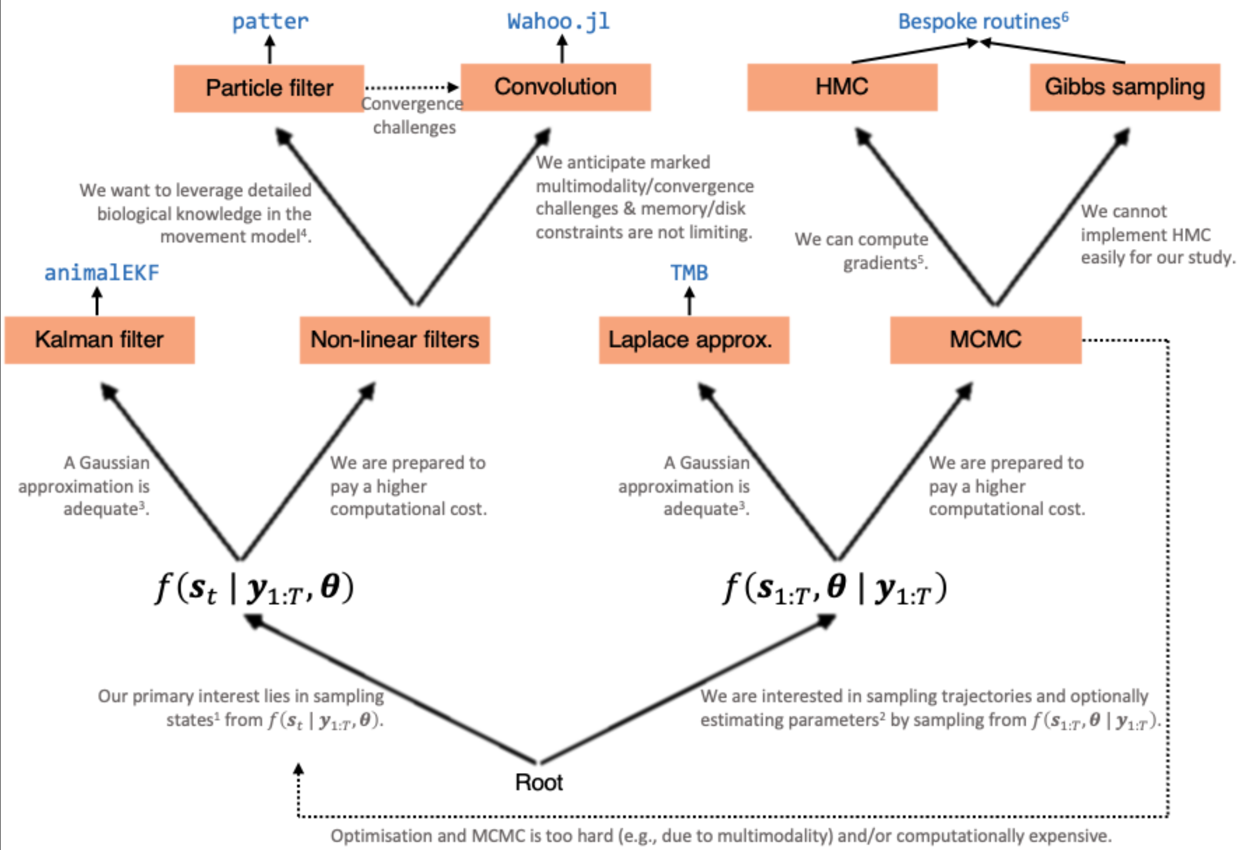
For inference, we can consider a selection of possible target distributions

State inference	State & parameter inference
$f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:t}, \boldsymbol{\theta})$	$f(\boldsymbol{s}_t, \boldsymbol{\theta} \mid \boldsymbol{y}_{1:t})$
$f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:T}, \boldsymbol{\theta})$	$f(\boldsymbol{s}_t, \boldsymbol{\theta} \mid \boldsymbol{y}_{1:T})$
$f(\boldsymbol{s}_{1:T} \mid \boldsymbol{y}_{1:T}, \boldsymbol{\theta})$	$f(\boldsymbol{s}_{1:T}, \boldsymbol{\theta} \mid \boldsymbol{y}_{1:T})$

Lavender et al. (in review)

Model-based inference

There are a number of inference algorithms available



Qualifications

[1] That is, we are primarily interested in 'snapshots' of the individual's state (e.g., for mapping space use or estimating residency), using existing knowledge of the movement and observation processes. We can extend routines on this side of the tree to sample parameters (from $f(\boldsymbol{s}_t, \boldsymbol{\theta} \mid \boldsymbol{y}_{1:T})$) or a few trajectories (from $f(\boldsymbol{s}_{1:T} \mid \boldsymbol{y}_{1:T}, \boldsymbol{\theta})$ or $f(\boldsymbol{s}_{1:T}, \boldsymbol{\theta} \mid \boldsymbol{y}_{1:T}, \boldsymbol{\theta})$) with increased computational cost, though bespoke routines may be required. For more comprehensive sampling of parameters and trajectories, MCMC algorithms may be the better choice.

[2] We should be prepared to rely on approximations and/or pay a higher computational price to sample from $f(\boldsymbol{s}_{1:T}, \boldsymbol{\theta} \mid \boldsymbol{y}_{1:T})$ compared to $f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:T}, \boldsymbol{\theta})$. An initial optimisation step may be required for efficient sampling.

[3] A Gaussian approximation may be appropriate if, for example, we assume a Gaussian random walk, a Gaussian detection probability model and our study system is far from movement/observation barriers (such as coastline).

[4] Currently, particle algorithms may be most appropriate option for practitioners, given the `patter` package and a compilation of examples for passive acoustic telemetry. Other options currently require statistical expertise to implement.

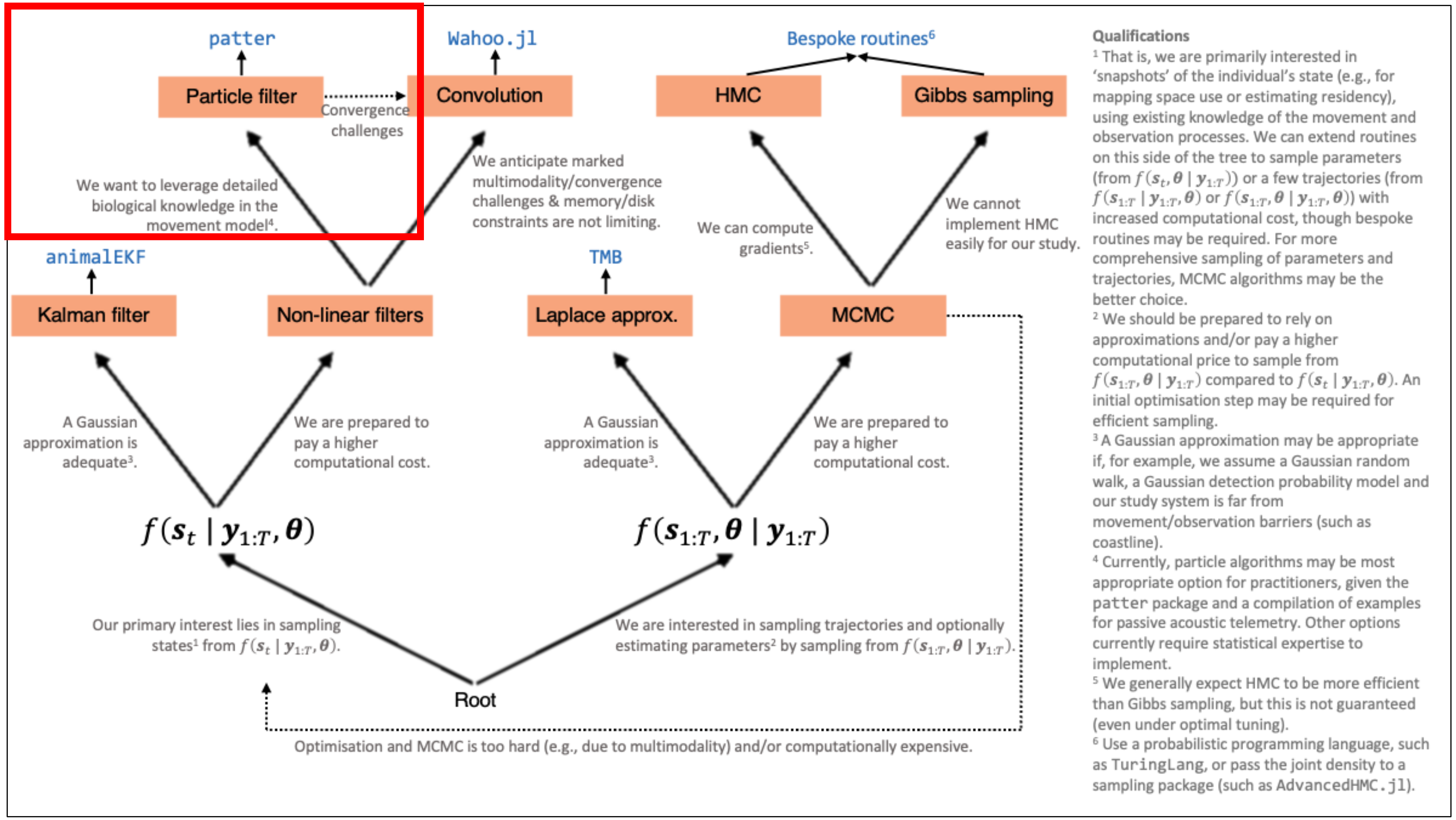
[5] We generally expect HMC to be more efficient than Gibbs sampling, but this is not guaranteed (even under optimal tuning).

[6] Use a probabilistic programming language, such as `TuringLang`, or pass the joint density to a sampling package (such as `AdvancedHMC.jl`).

Lavender et al. (in review).

Model-based inference

Particle algorithms are an obvious initial choice

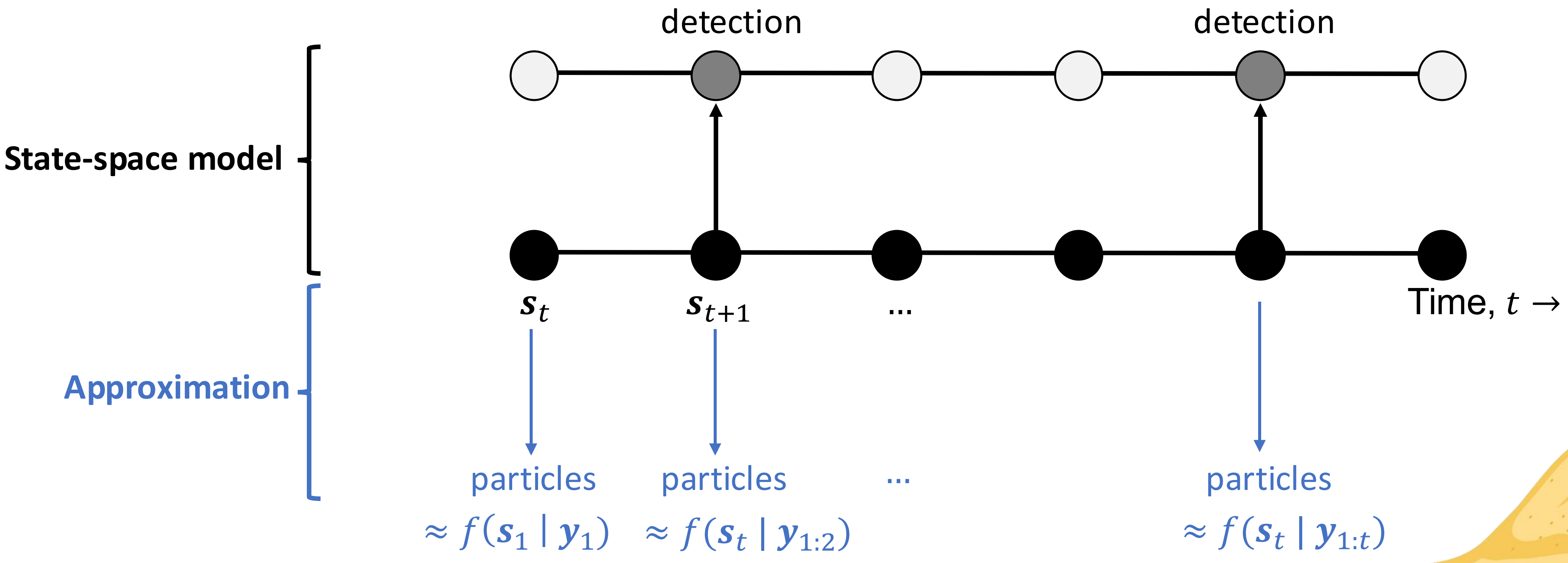


Lavender et al. (in review).

Particle algorithms

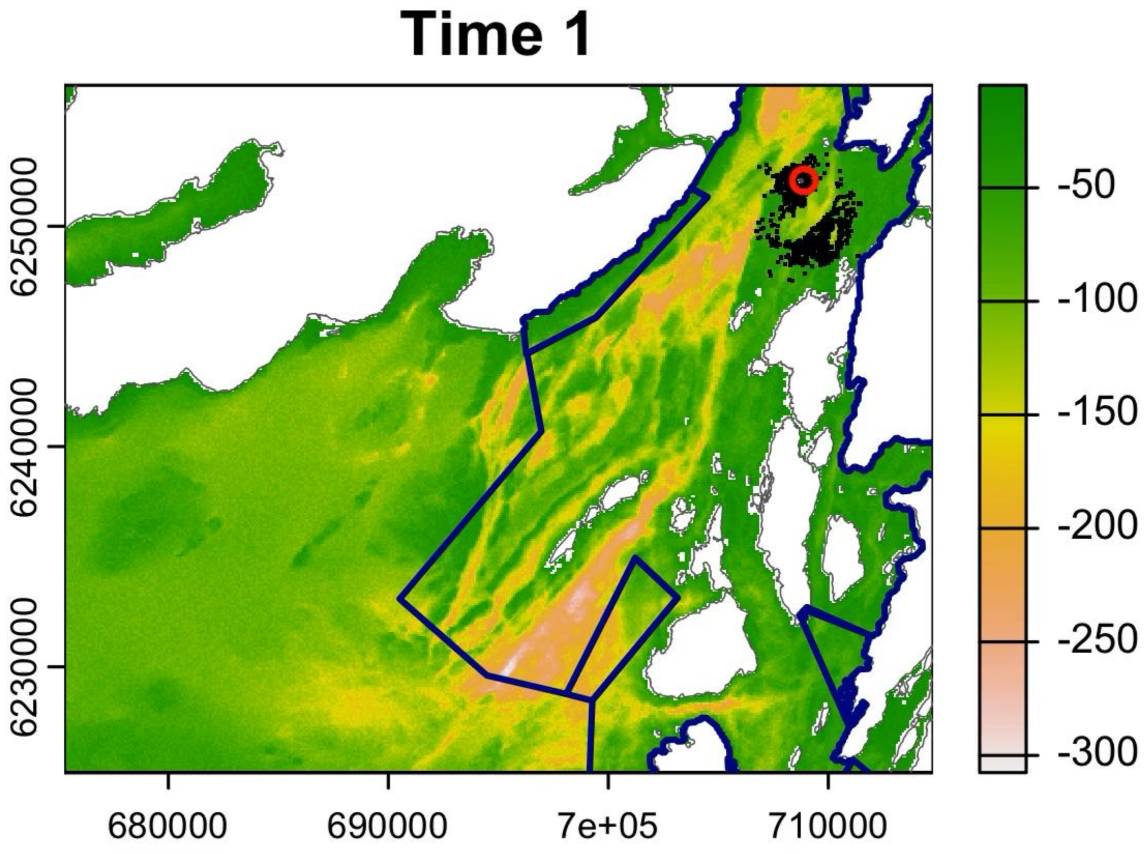
Particle algorithms

Particle filters are sequential MC algorithms that fit non-linear, non-Gaussian SSMs



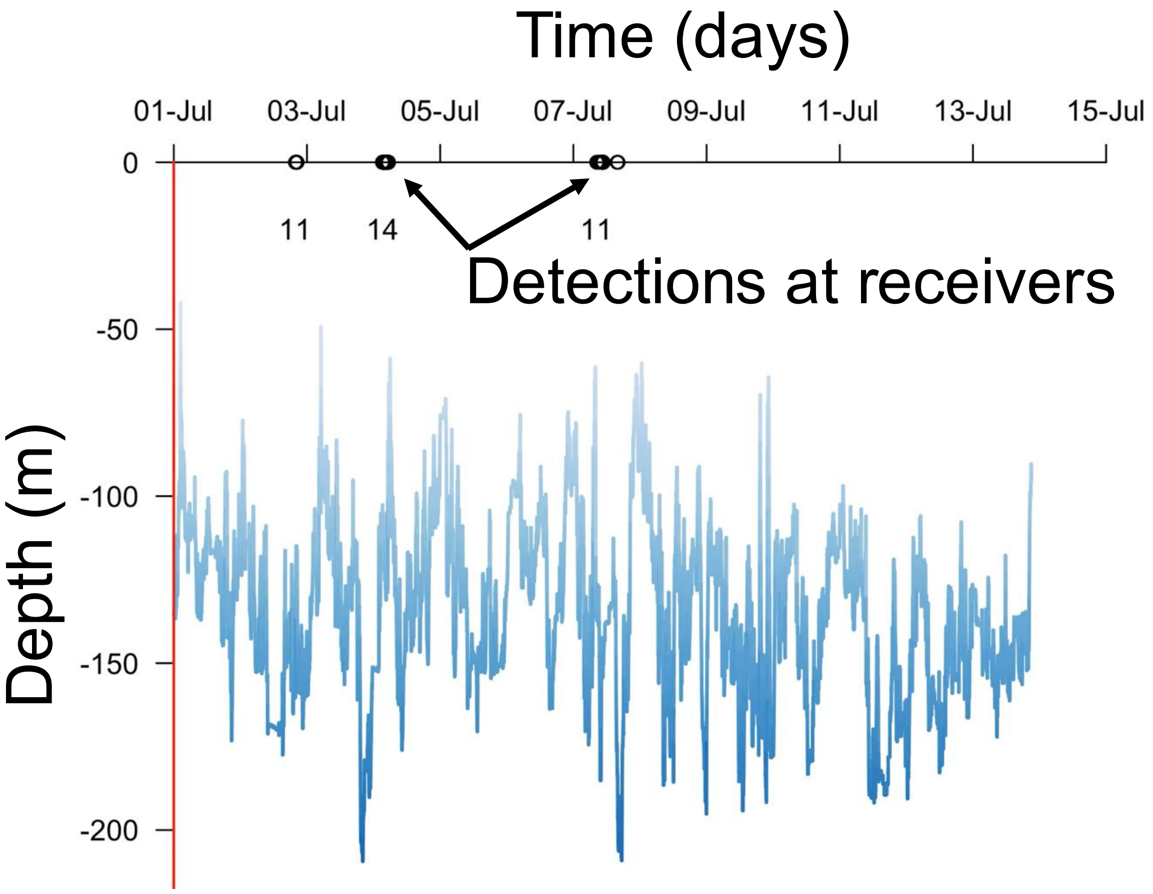
Particle algorithms

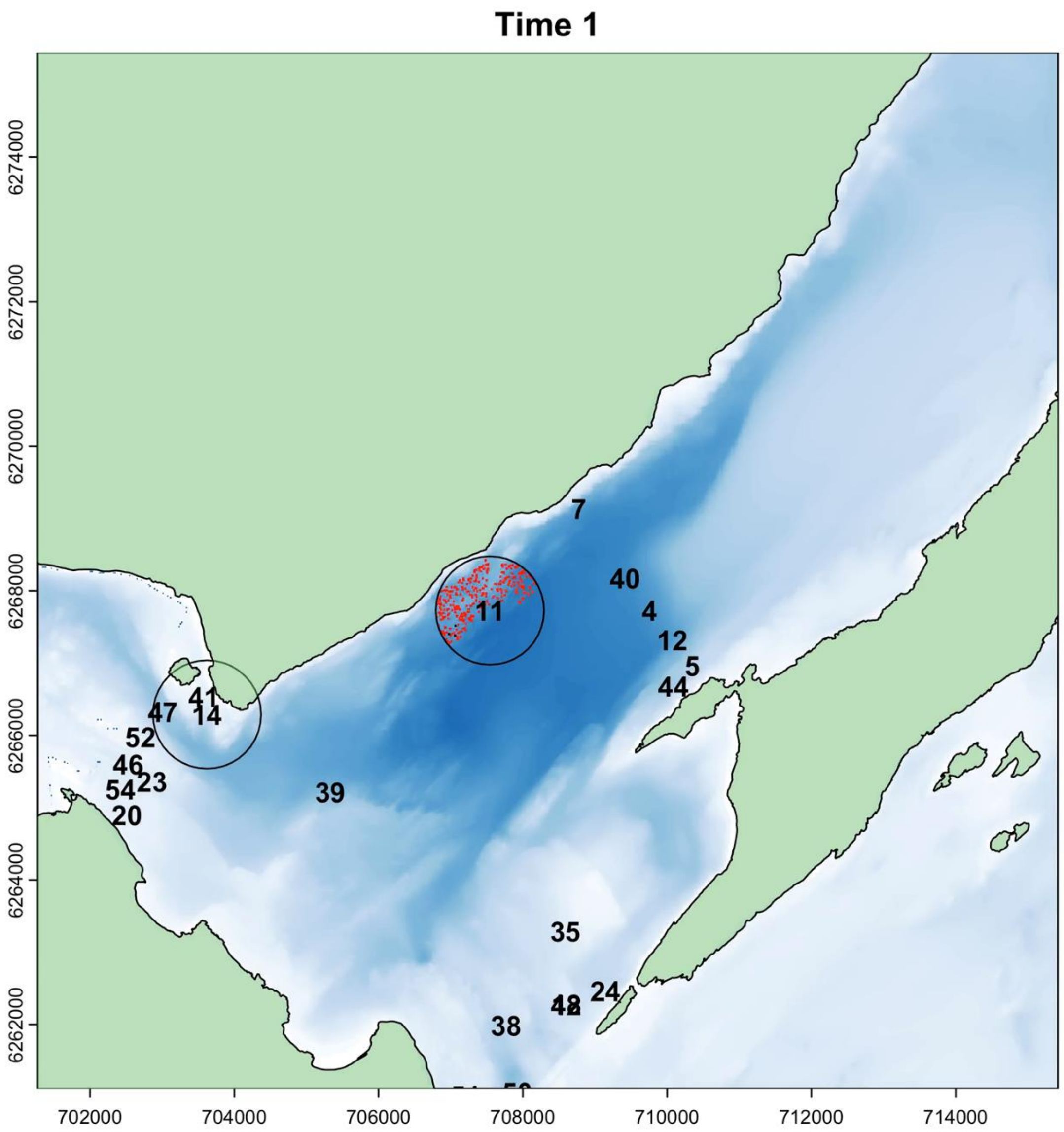
A simulation example of a particle filter in action



Particle algorithms

An example with real acoustic & archival data





Particle algorithms

Particle filtering algorithms target the partial marginal*, $f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:t})$

$$f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:t}) \propto \int f(\boldsymbol{s}_{t-1} \mid \boldsymbol{y}_{1:t-1})\, f(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1})\, d\boldsymbol{s}_{t-1}\, f(\boldsymbol{y}_t \mid \boldsymbol{s}_t)$$

'State' or location

(e.g., $\boldsymbol{s}_t = (x, y, \ldots)$)

Observations

- Acoustic data, $y_{k,t}^{(A)}$
- Depth data, $y_t^{(D)}$
- …

Movement process

- 2D RW
- 3D RW
- …

Observation process

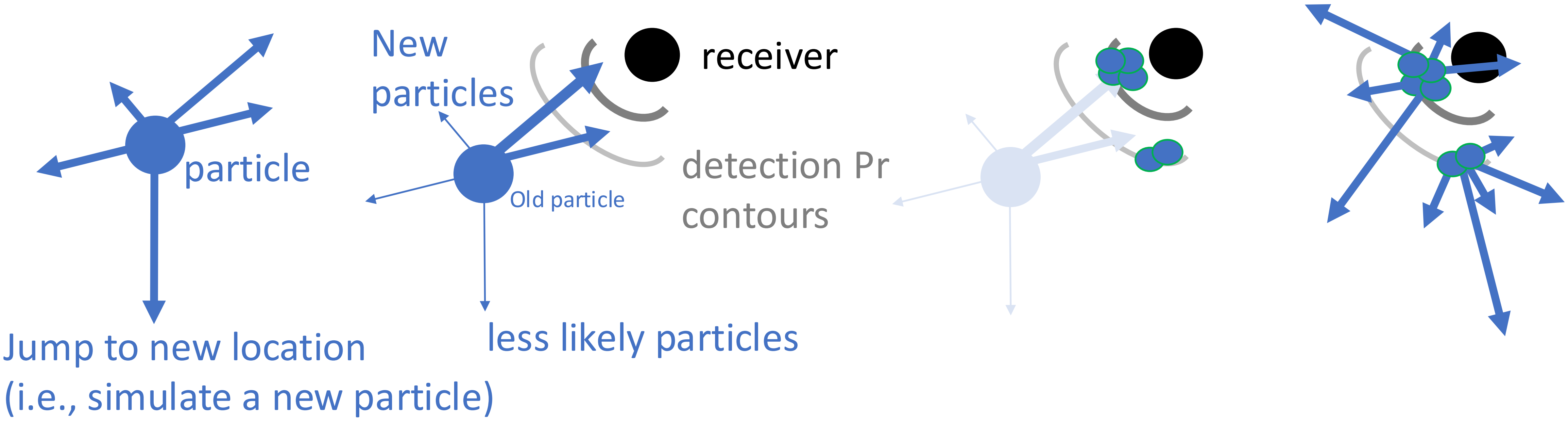
- $f\left(\boldsymbol{y}_t^{(A)} \,\middle|\, \boldsymbol{s}_t\right)$
- $f\left(y_t^{(D)} \,\middle|\, \boldsymbol{s}_t\right)$
- …

*We treat $\boldsymbol{\theta}$ as known. We can estimate $\boldsymbol{\theta}$ over multiple filter runs.

Particle algorithms

Particle filtering algorithms comprise four key steps

Simulation	Likelihood	Resampling	Iteration

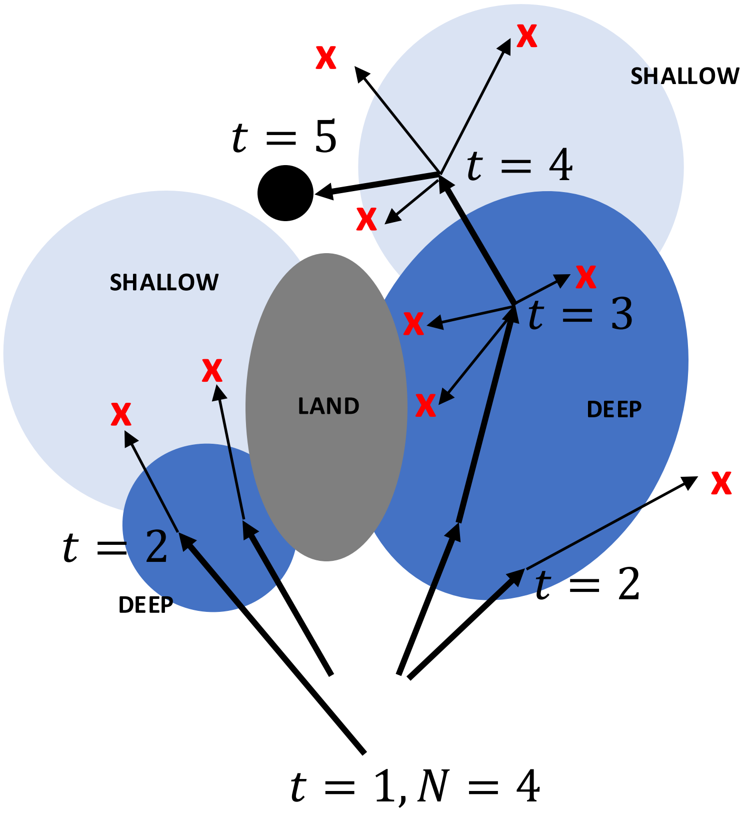


Particle algorithms

Particle filtering algorithms comprise four key steps

Time	Observations	
	$y_t^{(D)}$	$y_{t,k}^{(A)}$
1	-	0
2	DEEP	0
3	DEEP	0
4	SHALLOW	0
5	-	1

Data

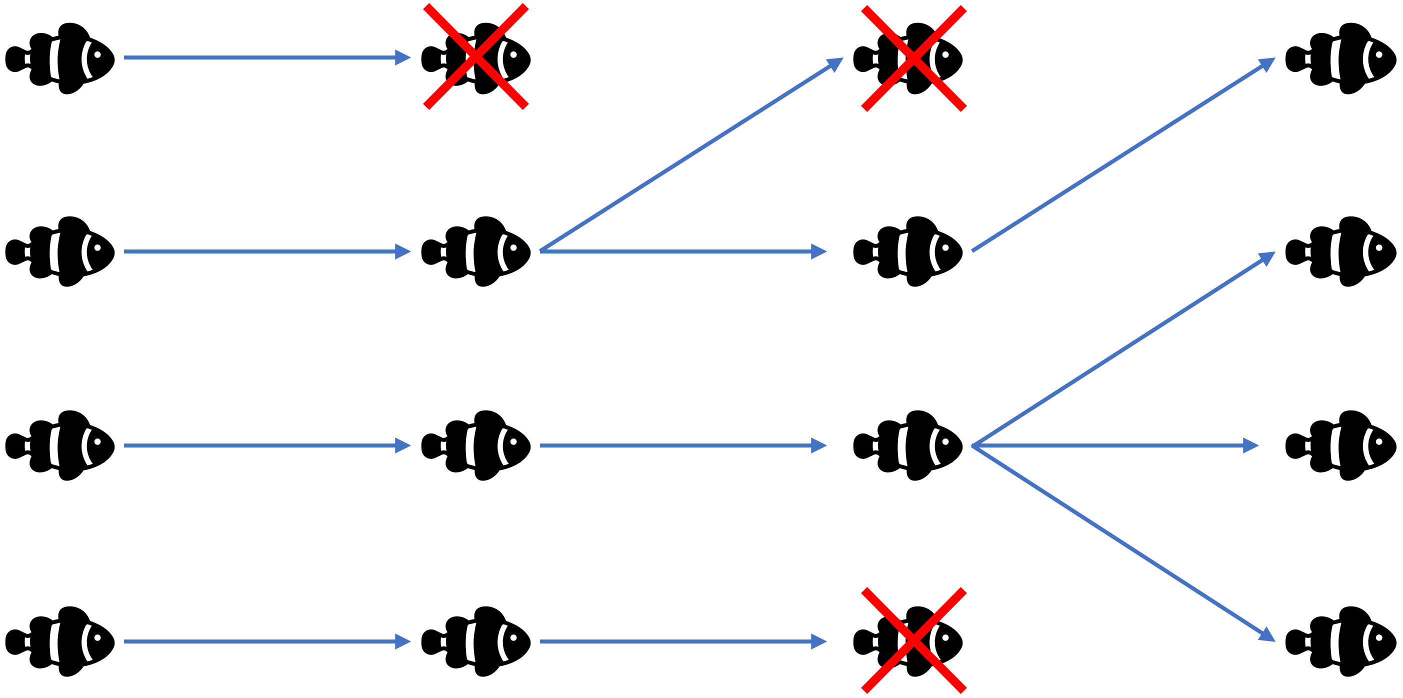


Filter, $f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:t})$

Samples from $f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:t})$ provide a 'snapshot' of the individual's location but not trajectories.

Particle algorithms

Particle algorithms are a blind, branching evolutionary (natural selection) process with no foresight



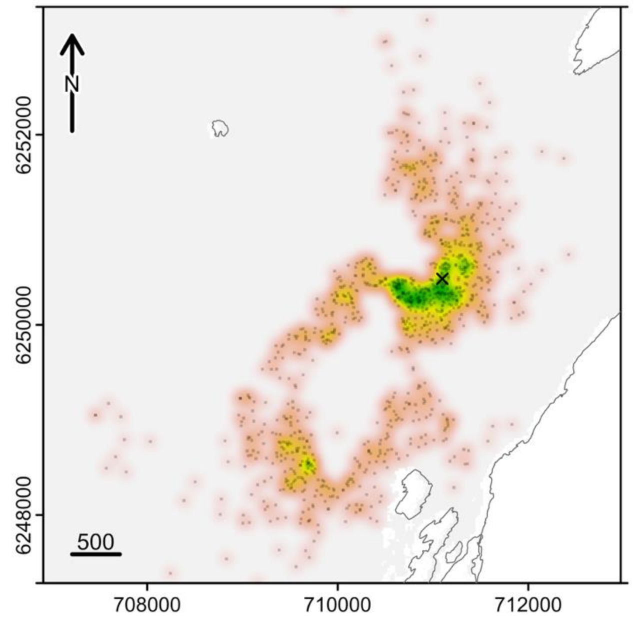
$f(\boldsymbol{s}_{t=1} \mid \boldsymbol{y}_{t=1})$ $f(\boldsymbol{s}_2 \mid \boldsymbol{y}_{1:2})$ $f(\boldsymbol{s}_3 \mid \boldsymbol{y}_{1:3})$ $f(\boldsymbol{s}_3 \mid \boldsymbol{y}_{1:4})$

https://methodsblog.com/2025/05/15/tracking-animals-with-particles/

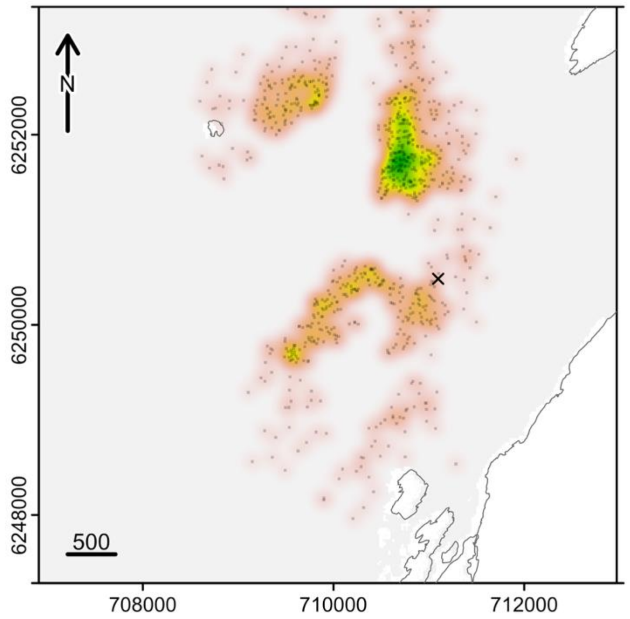
Particle algorithms

Particle smoothers approximate the full marginal, $f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:T})$

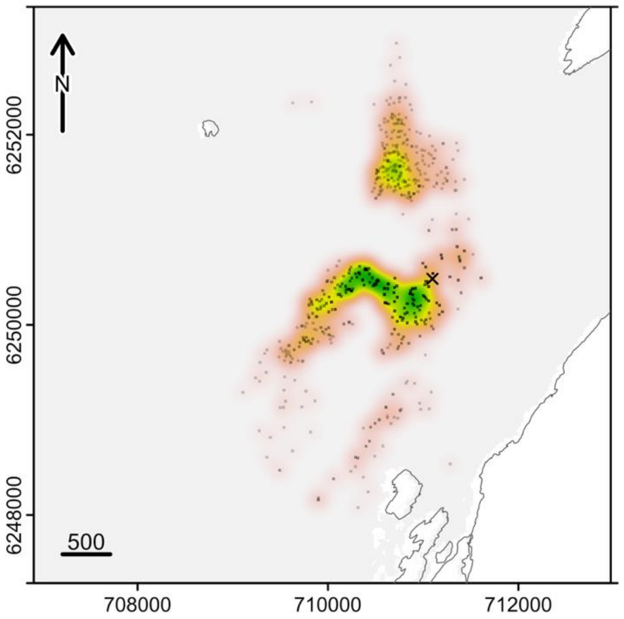
Forward filter



Backward filter



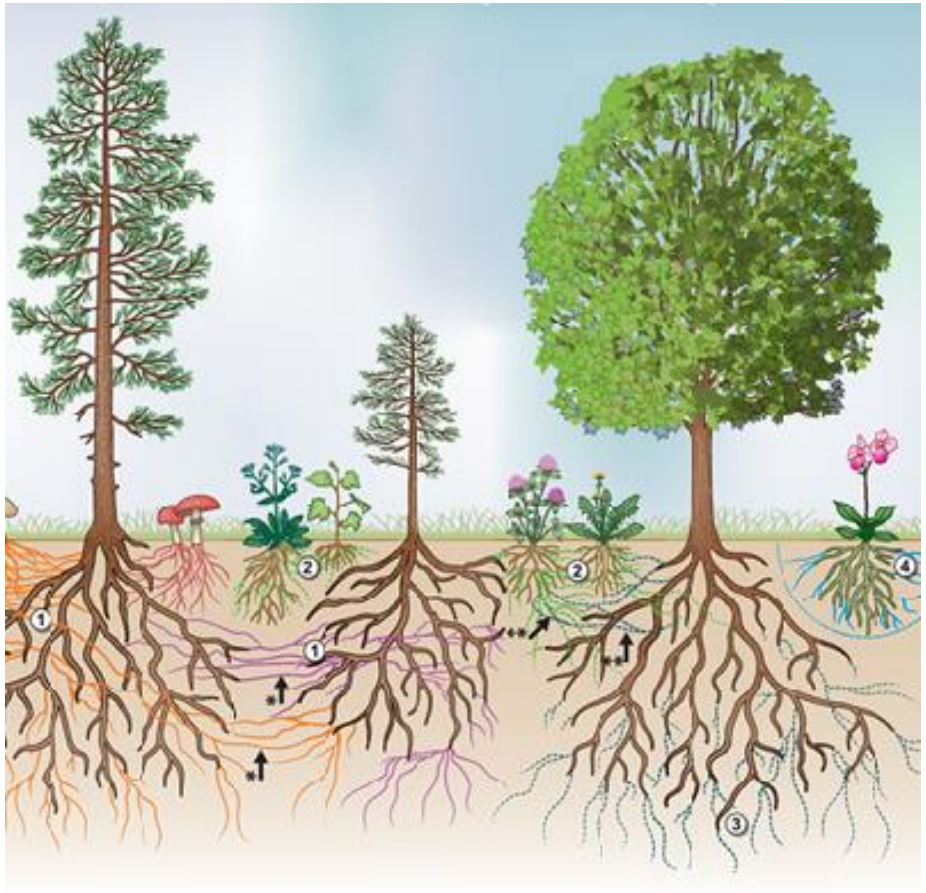
Smoother



If you know my location in five minutes time, you have more information about my location now.

Particle algorithms

Smoothing is like 'pruning'



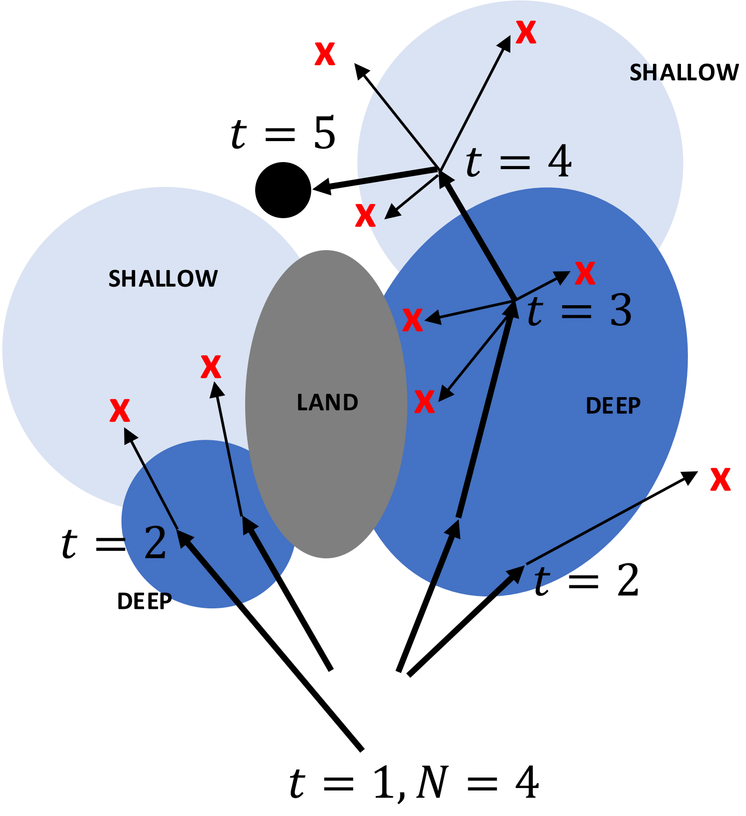
https://www.envision-dtp.org/2017/investigating-the-transfer-of-carbon-between-trees-via-common-mycorrhizal-networks/

Particle algorithms

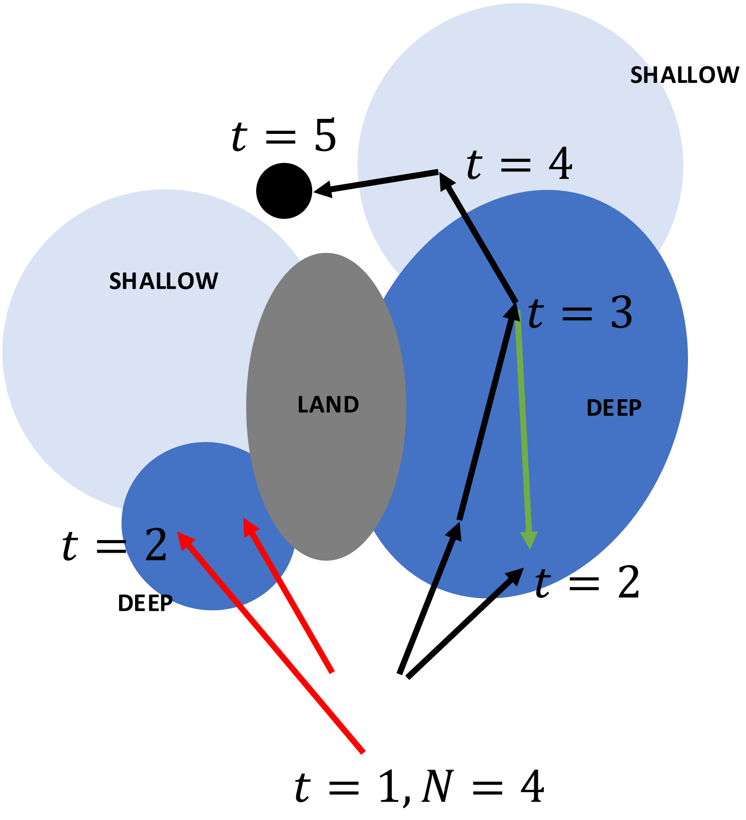
If you know my location in five minutes time, you have more information about my location now.

Time	Observations	
	$y_t^{(D)}$	$y_{t,k}^{(A)}$
1	-	0
2	DEEP	0
3	DEEP	0
4	SHALLOW	0
5	-	1

Data



Filter, $f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:t})$

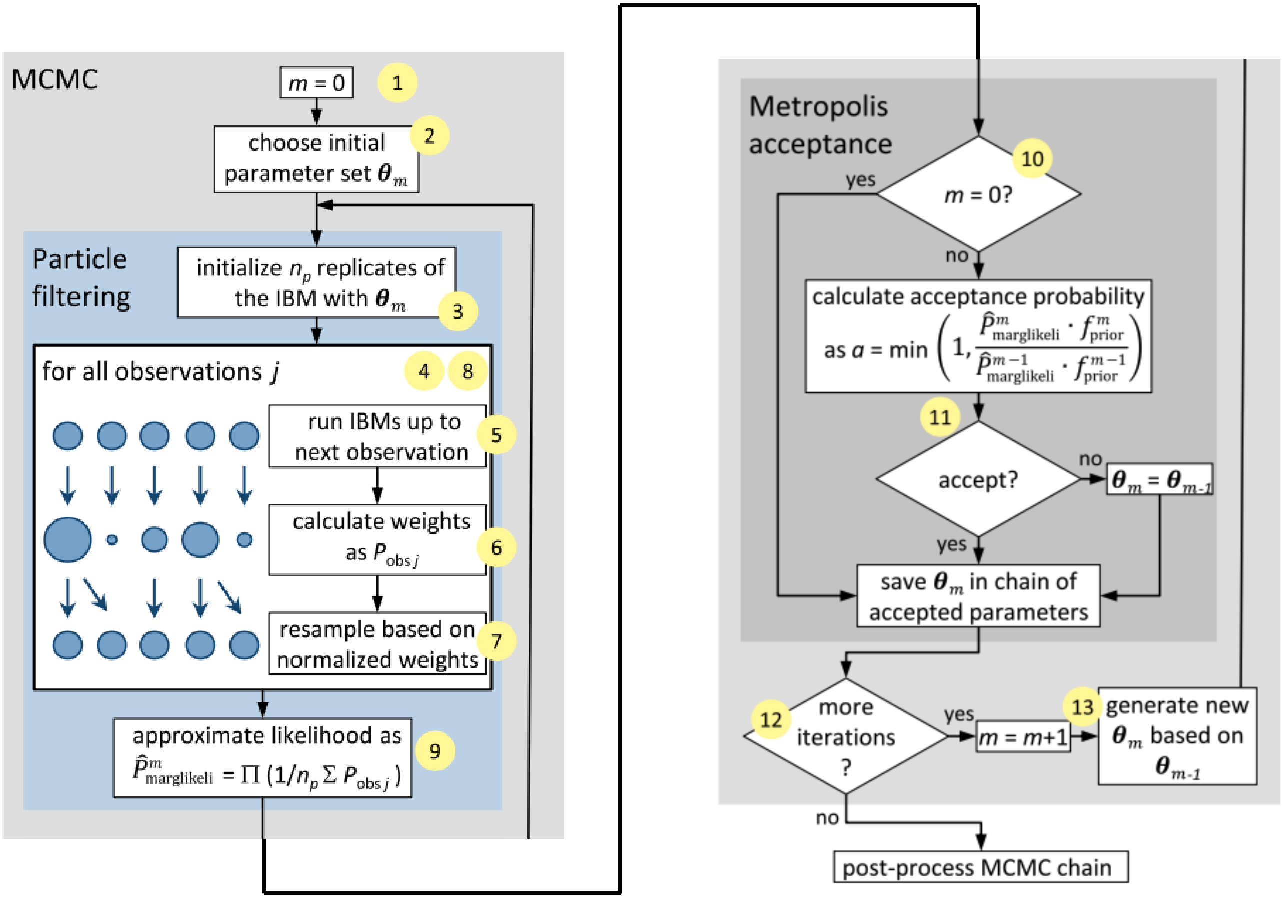


Smoother, $f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:T})$

Samples from $f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:t})$ or $f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:T})$ provide a 'snapshot' of the individual's location but not trajectories.

Particle algorithms

It is also possible to sample from $f(s_{1:T}, \theta \mid y_{1:T})$ but this can be expensive



Kattwinkel & Reichert (2017). Environ. Model. Softw.

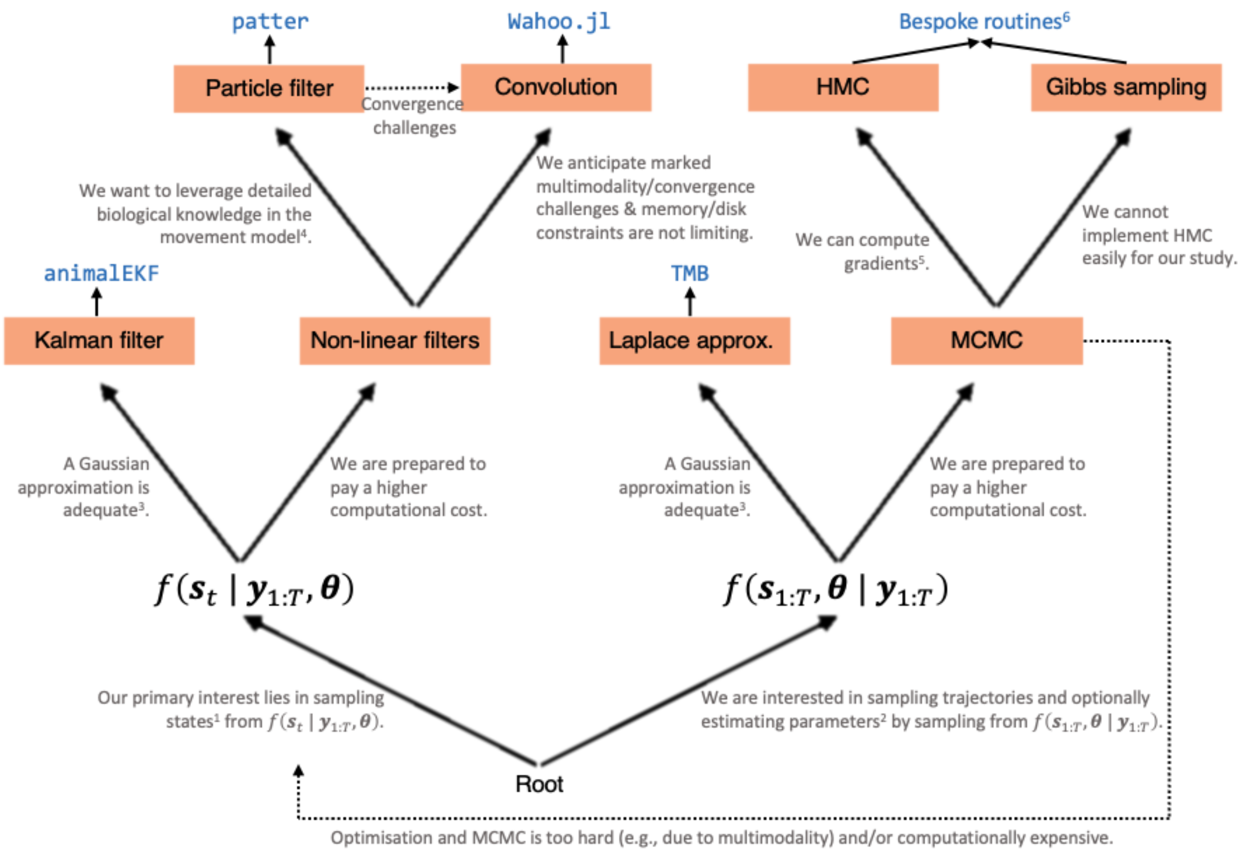
Particle algorithms

Particle algorithms have pros & cons compared to other model-based inference approaches

Pros	**Cons**
Flexibility (non-linear, non-Gaussian)	Particle smoothing is $\mathcal{O}(N^2T)$
Filtering is $\mathcal{O}(NT)$	Sampling trajectories &/or static parameters can be expensive
Accessibility via `patter`	Particle degeneracy

Particle algorithms

Particle algorithms have pros & cons compared to other model-based inference approaches



Qualifications

[1] That is, we are primarily interested in 'snapshots' of the individual's state (e.g., for mapping space use or estimating residency), using existing knowledge of the movement and observation processes. We can extend routines on this side of the tree to sample parameters (from $f(\boldsymbol{s}_t, \boldsymbol{\theta} \mid \boldsymbol{y}_{1:T})$) or a few trajectories (from $f(\boldsymbol{s}_{1:T} \mid \boldsymbol{y}_{1:T}, \boldsymbol{\theta})$ or $f(\boldsymbol{s}_{1:T}, \boldsymbol{\theta} \mid \boldsymbol{y}_{1:T}, \boldsymbol{\theta})$) with increased computational cost, though bespoke routines may be required. For more comprehensive sampling of parameters and trajectories, MCMC algorithms may be the better choice.

[2] We should be prepared to rely on approximations and/or pay a higher computational price to sample from $f(\boldsymbol{s}_{1:T}, \boldsymbol{\theta} \mid \boldsymbol{y}_{1:T})$ compared to $f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:T}, \boldsymbol{\theta})$. An initial optimisation step may be required for efficient sampling.

[3] A Gaussian approximation may be appropriate if, for example, we assume a Gaussian random walk, a Gaussian detection probability model and our study system is far from movement/observation barriers (such as coastline).

[4] Currently, particle algorithms may be most appropriate option for practitioners, given the `patter` package and a compilation of examples for passive acoustic telemetry. Other options currently require statistical expertise to implement.

[5] We generally expect HMC to be more efficient than Gibbs sampling, but this is not guaranteed (even under optimal tuning).

[6] Use a probabilistic programming language, such as `TuringLang`, or pass the joint density to a sampling package (such as `AdvancedHMC.jl`).

Lavender et al. (in review)

patter

patter

The patter R package implements particle algorithms

`patter`

Particle algorithms for animal movement modelling in `R`

repo status Active | lifecycle experimental | CRAN not published | coverage 83% | R-CMD-check passing

`patter` provides particle filtering, smoothing and sampling algorithms for animal movement modelling, with a focus on passive acoustic telemetry systems. This wraps and enhances a fast `Julia` backend (`Patter.jl`). The methodology enables the reconstruction of movement paths and patterns of space use. `patter` unifies a suite of methods formerly known as the `flapper` algorithms and supersedes the experimental `flapper` package (Lavender et al., 2023).

> **Note:** `patter` is a new `R` package. Like all new packages, you should use it with a degree of caution. Please share feedback and issues.

> **NEWS** Welcome to `patter v.2.0.0`! This includes some **breaking changes**. For projects based on earlier versions, use `renv`. For future projects, `patter v.2.0.0` is recommended.

Lavender et al. (2025). Meth. Ecol. Evol.

github.com/**edwardlavender/patter**

patter

patter wraps a high-performance Julia package, Patter.jl

Patter.jl

Particle algorithms for animal movement modelling in Julia

repo status Active | Documenter passing | Runtests passing | codecov unknown

`Patter.jl` provides particle filtering, smoothing and sampling algorithms for animal movement modelling, with a focus on passive acoustic telemetry systems. The package is heavily based on the `ParticleFish` package developed by Andreas Scheidegger. `Patter.jl` forms the backend for the `patter` `R` package.

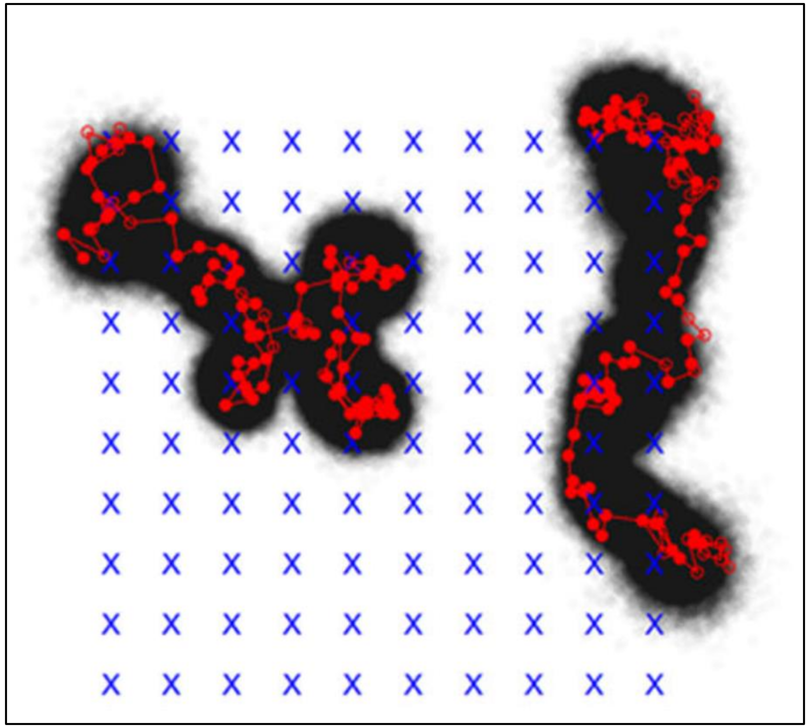
> **Note:** `Patter.jl` is a new `Julia` package. Like all new packages, you should use it with a degree of caution. Please share feedback and issues.

Particle algorithms: implementation

Packages are easy-to-use, flexible & much faster than related routines

Packages support:

- Multi-dimensional states
- Custom movement models
- Diverse animal tracking datasets
- Multi-threading

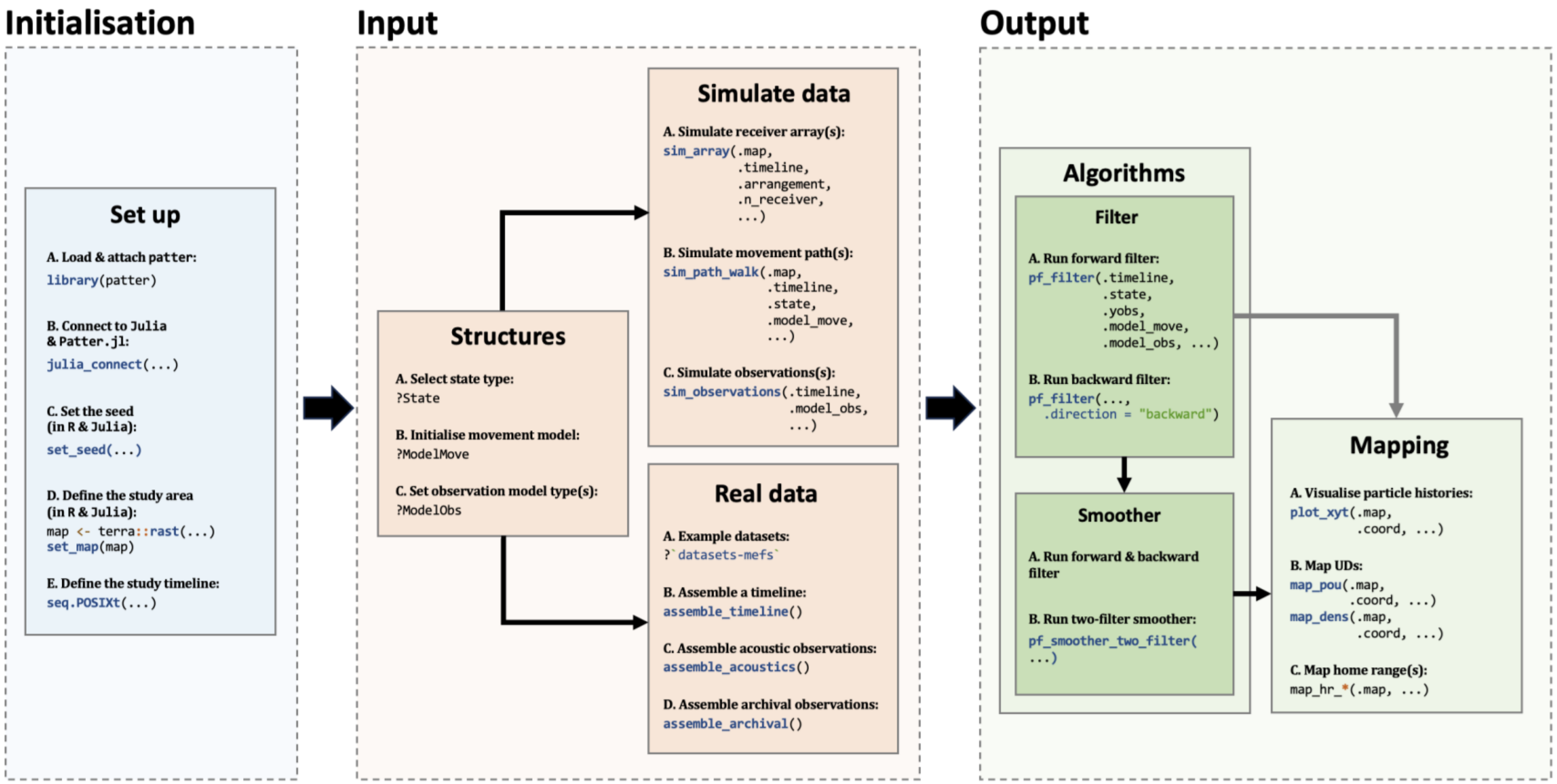


cf. Hostetter & Royle (2020)

Lavender et al. (2025). Meth. Ecol. Evol.

patter

Packages are easy-to-use, flexible & much faster than related routines



Lavender et al. (2025). Meth. Ecol. Evol.

`patter`

We recently released v.2.0.0 of the packages

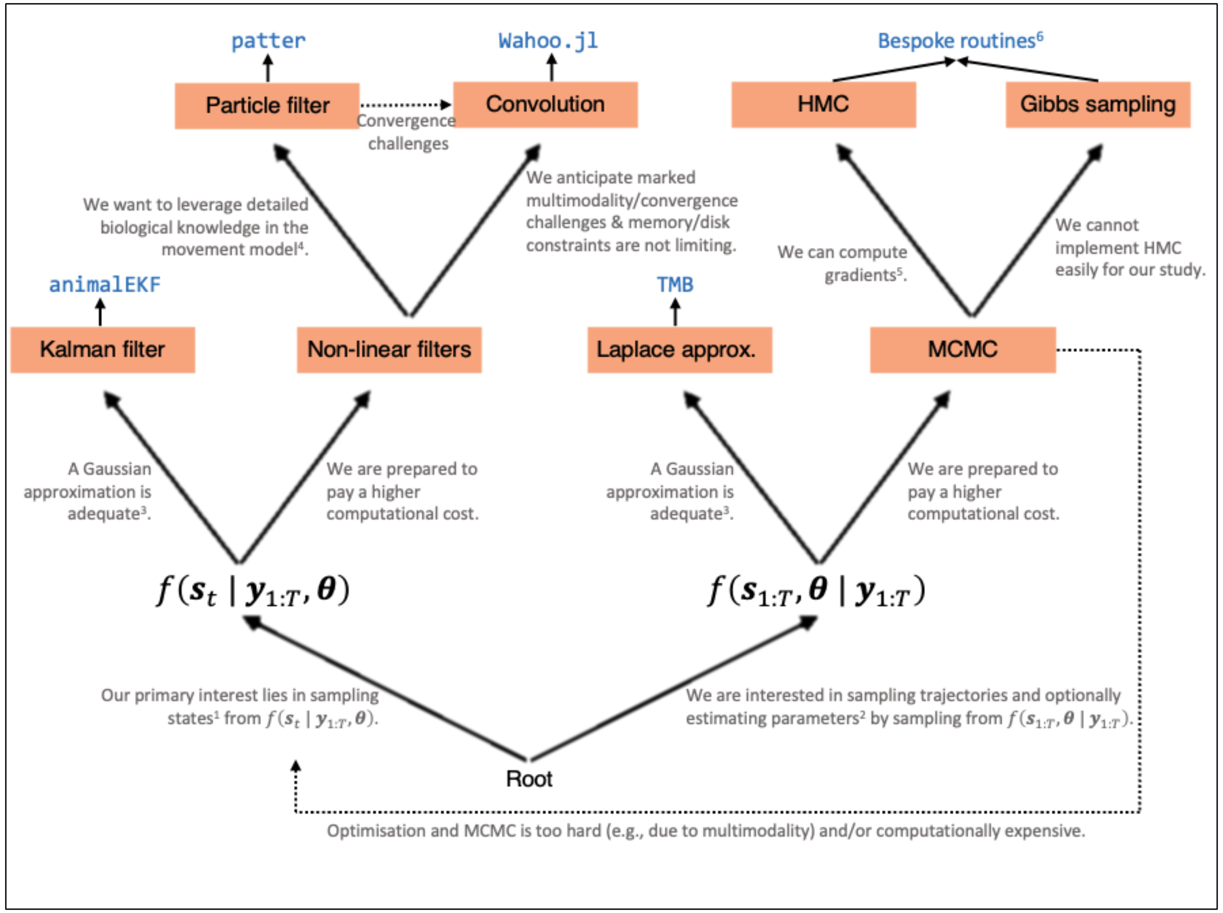
> **NEWS** Welcome to `patter v.2.0.0`! This includes some **breaking changes**. For projects based on earlier versions, use `renv`. For future projects, `patter v.2.0.0` is recommended.

For support, see:

- https://github.com/edwardlavender/patter
- https://github.com/edwardlavender/patter/issues
- https://github.com/edwardlavender/patter-workshops

Ongoing developments

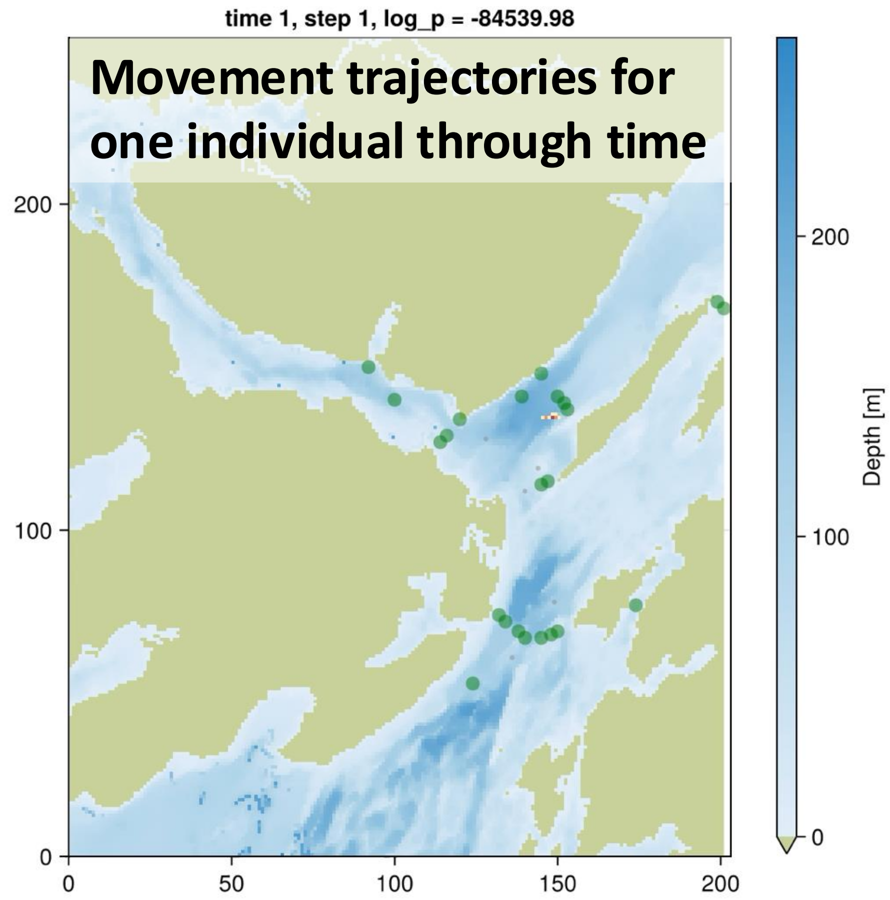
- In `patter`, we are moving to a phase of **community-driven** feedback
- A separate **`patter.workflows`** package is available
- We are developing other model-based inference algorithms & packages, such as **`Wahoo.jl`**, for animal tracking



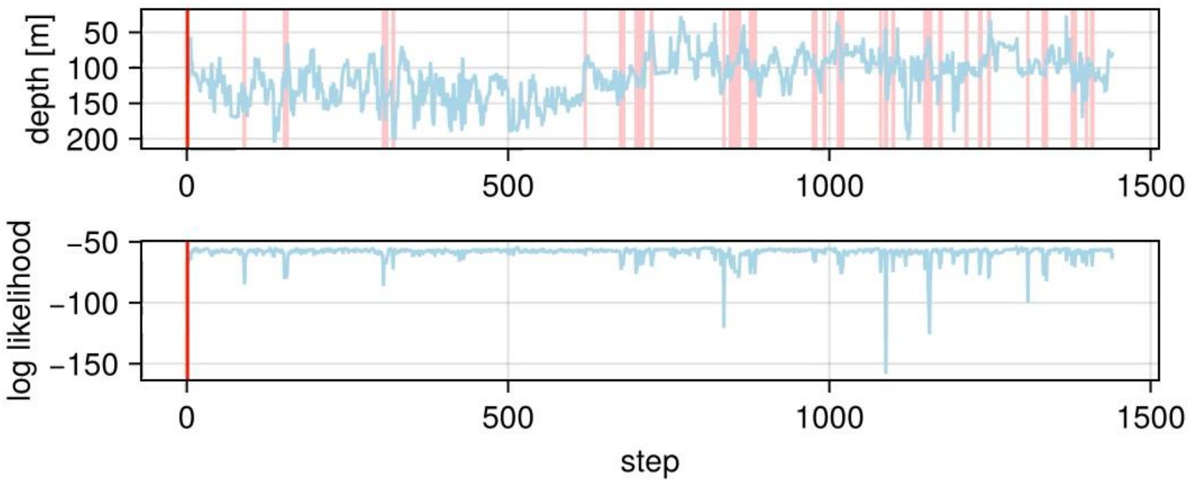
Lavender et al. (in review)

Ongoing developments

Wahoo.jl implements filtering without sampling plus smoothing & trajectory inference, leveraging GPU-acceleration



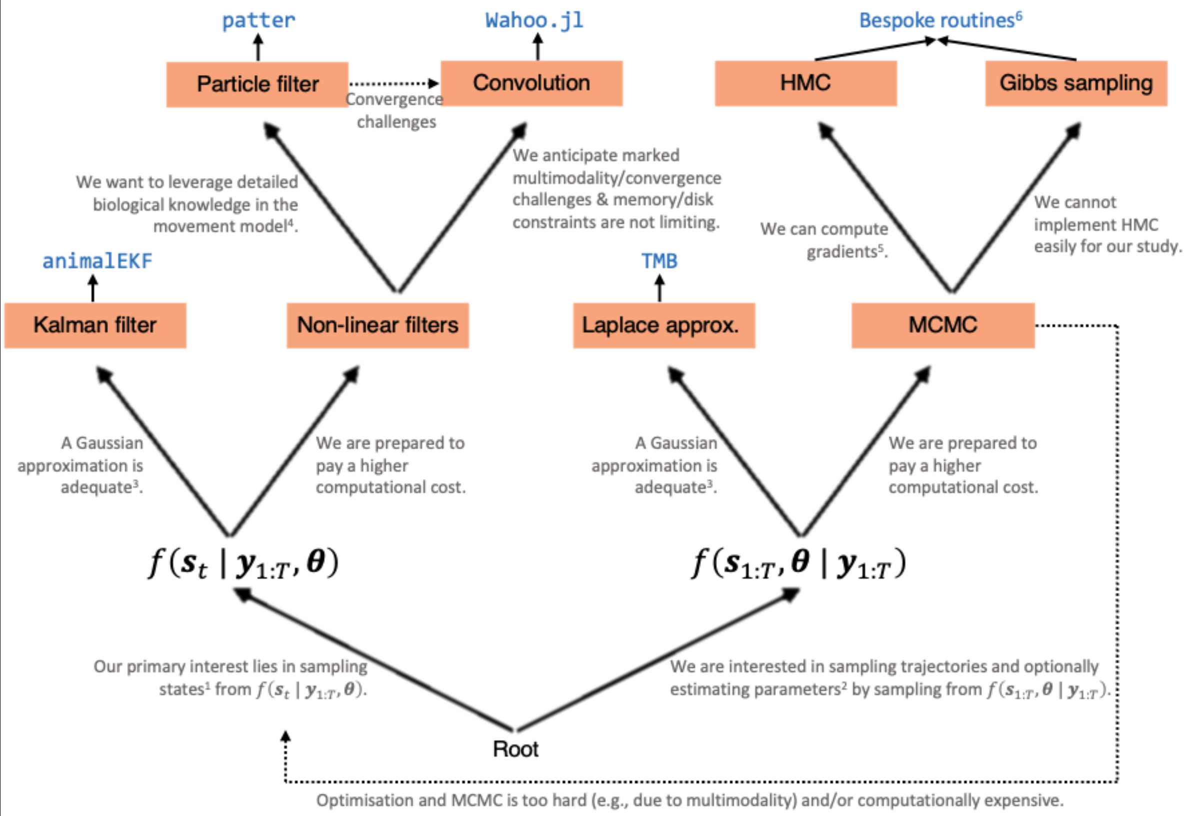
Movement trajectories for one individual through time



https://github.com/scheidan/**Wahoo.jl**

Ongoing developments

There is still much work to be done to develop model-based inference routines for animal tracking



Qualifications

[1] That is, we are primarily interested in 'snapshots' of the individual's state (e.g., for mapping space use or estimating residency), using existing knowledge of the movement and observation processes. We can extend routines on this side of the tree to sample parameters (from $f(\boldsymbol{s}_t, \boldsymbol{\theta} \mid \boldsymbol{y}_{1:T})$) or a few trajectories (from $f(\boldsymbol{s}_{1:T} \mid \boldsymbol{y}_{1:T}, \boldsymbol{\theta})$ or $f(\boldsymbol{s}_{1:T}, \boldsymbol{\theta} \mid \boldsymbol{y}_{1:T}, \boldsymbol{\theta})$) with increased computational cost, though bespoke routines may be required. For more comprehensive sampling of parameters and trajectories, MCMC algorithms may be the better choice.

[2] We should be prepared to rely on approximations and/or pay a higher computational price to sample from $f(\boldsymbol{s}_{1:T}, \boldsymbol{\theta} \mid \boldsymbol{y}_{1:T})$ compared to $f(\boldsymbol{s}_t \mid \boldsymbol{y}_{1:T}, \boldsymbol{\theta})$. An initial optimisation step may be required for efficient sampling.

[3] A Gaussian approximation may be appropriate if, for example, we assume a Gaussian random walk, a Gaussian detection probability model and our study system is far from movement/observation barriers (such as coastline).

[4] Currently, particle algorithms may be most appropriate option for practitioners, given the `patter` package and a compilation of examples for passive acoustic telemetry. Other options currently require statistical expertise to implement.

[5] We generally expect HMC to be more efficient than Gibbs sampling, but this is not guaranteed (even under optimal tuning).

[6] Use a probabilistic programming language, such as `TuringLang`, or pass the joint density to a sampling package (such as `AdvancedHMC.jl`).

Lavender et al. (in review)

Resources

For the particle algorithm methodology, see:
Lavender, E. et al. (2025a). Particle algorithms for animal movement modelling in receiver arrays. Meth. Ecol. Evol. https://doi.org/10.1111/2041-210X.70028

For the `patter` package, see:
Lavender, E. et al. (2025b). patter: particle algorithms for animal tracking in R & Julia. Meth. Ecol. Evol. https://doi.org/10.1111/2041-210X.70029

For a case-study analysis, see:
Lavender, E. et al. (in review). Animal tracking with particle algorithms for conservation. bioRxiv. https://doi.org/10.1101/2025.02.13.638042

For the wider picture, see:
Lavender, E. et al. (in review). State-space models and inference approaches for aquatic animal tracking with passive acoustic telemetry and biologging sensors. EcoEvoRxiv. https://doi.org/10.32942/X2MP84

Resources

For the `patter` package, see:
[https://github.com/edwardlavender/**patter**](https://github.com/edwardlavender/patter)

For workshop materials, see:
[https://github.com/edwardlavender/**patter-workshops**](https://github.com/edwardlavender/patter-workshops)

For `Wahoo.jl`, see:
[https://github.com/scheidan/**Wahoo.jl**](https://github.com/scheidan/Wahoo.jl)

Contact

edward.lavender@eawag.ch

edwardlavender

With thanks to Helen Moor, Andreas Scheidegger, Carlo Albert, Stanisław Biber, Sophie Smout, Janine Illian, James Thorburn and many others.

Photograph courtesy of James Thorburn