# The Statistical Building Blocks of Animal Movement Simulations

Wayne M. Getz*[1,2,3], Richard Salter*[3,4], Varun Sethi[1], Shlomo Cain[5], Ran Nathan[6], Orr Spiegel[5], and Sivan Toledo[7]

[1]*Department Environmental Science, Policy and Management, University of California, Berkeley, 94720, CA, USA*
[2]*School of Mathematics, Statistics & Computer Science, University of KwaZulu-Natal, South Africa*
[3]*Numerus Inc., 850 Iron Point Road, Folsom, CA 95630, USA*
[4]*Department of Computer Science, Oberlin College, OH 44074, USA*
[5]*School of Zoology, Faculty of Life Sciences, Tel Aviv University, Tel Aviv 69978, Israel*
[6]*Department of Ecology, Evolution and Behavior, The Hebrew University of Jerusalem, Jerusalem, Israel*
[7]*Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel*

**Abstract**

Animal movement plays a key role in many ecological processes and has a direct influence on the individual's fitness. Movement tracks can be studied at various spatio-temporal scales, but the movement usually emerges from sequences of fundamental movement elements (FuMEs: e.g., a step or wing flap). The importance of movement in all aspects of the life history of mobile animals highlights the need to develop tools for generating realistic tracks. These individual movement elements are typically not extractable from relocation tracking data: a time series of sequential locations (e.g., GPS fixes) from which we can extract step-size or length (SL, aka velocity when sampling frequency is fixed) and turning-angle (TA) time series. The statistics of these extracted quantities (means, standard deviations, correlations), computed for fixed short segments of track (e.g., 10-30 point sequences), can then be used to categorize such segments into statistical movement elements (StaMEs; e.g. directed fast movement versus random slow movement elements). StaMEs from the same category can be used as a basis for constructing step-selection kernels of a particular movement type, referred to as a canonical activity mode (CAM; e.g. foraging, or commuting). Such CAM kernels can be used to simulate tracks from pure or characteristic mixtures of two or more CAMs, where such mixtures are recognized as a particular type of activity: e.g., foraging or directed walking with interruptions for vigilance or navigation bouts. These activity types in turn (e.g., interspersed sequences of resting, foraging, and directed movements) can then be organized into temporally fixed (24-hour) diel activity routines (DARs; e.g., a day of migration vs a day of local movement). We suggest that a simulator accounting specifically for the hierarchical nature of movement able to generate segments of tracks that may fit CAM length sections of real movement data. Hence, here we present a two-mode, step-selection kernel simulator (Numerus ANIMOVER_1), built using Numerus RAMP technology. We discuss methods for selecting kernel parameters that generate StaMEs comparable to those observed in empirical data, thereby providing a general tool for simulating animal movement in diverse contexts such as evaluation on animal response to landscape changes. We illustrate our methods for extracting StaMEs from both simulated and real data—in our case the latter is movement data of two barn owls (*Tyto alba*) in the Harod Valley, Israel. The methods described here may provide a way to fit fine scale simulations to empirical data to in ways that may allow us to identify stressed or diseased individuals.

**Keywords**

Multiscale movement analysis, movement path segmentation, diel activity routine (DAR), **Numerus ANIMOVER**, Barn owl *Tyto alba*, hierarchical cluster analysis

*Co-first authors

# 1 Introduction

One of the major challenges common to several subfields of ecology (e.g., conservation biology, disease ecology, resource ecology) is predicting how the movement of animals changes in response to landscape factors and the state or health of the individual [1, 2, 3]. Examples of change include the spatio-temporal distribution of resources vital to the existence of individuals [4, 3], the movement of animals released into new surroundings for the purposes of conservation [5, 2], or the movement of individuals under stress or with infections [6, 7]. The dynamic resource example has led to the concept of resource tracking, which has become an active field of research in movement ecology [8, 9, 10]. Quantification of this process through models that link animal movement to resources will enable us to better predict the consequences of global change on animal populations [11]. The stressed individual example may be critical to the continued existence of endangered species [12]. The third example may help us identify and selectively remove individuals that are sick, and hence reduce the risk of pandemic outbreaks [13].

Movement, whether simulated or real, generates complex patterns that require various approaches to classify and comprehend. One such approach is to break down movement tracks into hierarchical components that are relevant at different spatiotemporal scales of analysis [14, 15, 16, 17] (also see Appendix A.1). The value of an hierarchical approach is abundantly evident as an epistemological tool for deconstructing complexity (e.g., the construction of texts and genomes), but requires some building-block basis (e.g., letters for texts, codons for genetic coding of proteins), whether real or imagined, for the hierarchical construction [18]. In the context of movement tracks, the real building blocks are fundamental movement elements (FuMEs; [15]—e.g., for horses this may be walk, trot, cantor and gallop, etc.), but these usually cannot be extracted from relocation data, even when such data have a resolution on the order of seconds. Identification of the FuME sequence as a animal moves through space is likely to require either analyses of videos of the movements, inferences made from data using accelerometers placed at particular locations on the animals body require careful protocals [19], or other types of data collected from sufficiently fast sensors to identify the start and end of each type of movement element (e.g. a wingbeat of a bird [20]). In the absence of being able to identify the real FuME sequence, we propose the construction of statistical movement elements as the smallest achievable building block elements for the hierarchical construction of animal movement tracks.

Empirically, the movement track of an individual over a landscape is generally represented by a sequence of locations that is recorded using GPS technology [21], ATLAS reverse GPS technology [22], acoustic receivers or other technologies [23]. From such sets of relocation points, also referred to as a "walk," step-length (SL; also velocities when the sampling frequency is fixed) and turning-angle (TA) time series can be extracted [15] (Appendix A.2). These time series can then be used to compute various derived quantities, such as radial and tangential velocities at each relocation point, and auto correlations of variables along segments of the movement track [24]. The statistics of such variables, computed for fixed short segments of track (e.g., 10-30 points), can then be used to categorize such segments into statistical movement elements, which we here refer to as StaMEs (previously called metaFuMEs, as in [15]).

These StaMEs can then be classified into a limited number of categories, as demonstrated in this paper (e.g. a short element of direct fast flight, or one of a tortuous slow walk). A string of same category StaMEs then constitutes a track segment that can be classified as a homogeneous or canonical activity mode (CAM) of a type defined by the underlying category of StaME (e.g., running, walking in a persistent direction, flitting around in a somewhat random manner). Characteristic mixtures of CAMs, in turn, can be strung together as identifiable activity types (e.g., resting, foraging, heading to a known location while being vigilant), with several different activity types coming together each day to form a diel activity routine (DAR) [25, 26]. The DAR itself is a hierarchical segment that can be understood in terms of an invariant 24-hour period for most animals, apart from some deep dwelling marine or cave-dwelling species, because for most species it is a fundamental biological rhythm honed by evolution [27]. The periods of various exogenous environmental cycles beyond the diel period (e.g., lunar or seasonal), though, can be quite variable in their effects on species, depending on the latitude [28], elevation [29] and the trophic levels (e.g., herbivore, predator, scavenger) at which they function.

If the relocation sampling frequency is relatively high (i.e., $\approx 5$ or more relocation points per min), then the statistical properties of a segment of, say, 10-30 consecutive points (e.g., the means of the velocities and others) can serve to construct a set of statistical movement element (StaMEs) segments, which may then be classified into a relatively small set of StaME categories and associated canonical activity modes. From our presentation here, it will become clear that StaMEs are dependent, firstly, on the resolution (i.e., frequency) of the relocation data and, secondly,

on the number of consecutive points used to derive the statistics of our StaMEs (i.e., its duration). Since some of the measures and features used to characterize movement track segments using relocation data are noticeably frequency dependent [30, 31], these will be influenced by the scale at which we define the underlying set of StaMEs used to reconstruct movement track segments of various lengths.

In this paper we also propose a tool to generate realistic synthetic animal trajectories. The methods that we develop below provide a way to fit fine scale simulation data to movement in general or, in particular, predicting changes in the movement behavior of animals or identify animals under stress. The latter may help us to predict how an individual's movement through space may respond to global change, or to identify diseased or infected individuals. For clarity of exposition, we first present a mathematical formulation of our model framework. When then discuss our Numerus RAMP (i.e., runtime alterable modeling platform) implementation of our model, which we name ANIMOVER_1 (**ANI**mal **MOVE**ment **R**AMP; the "1" anticipates future elaborated versions). We discuss why the construction of such RAMPs can help facilitate analyses in ways that are more difficult using current programming platforms, such as Netlogo [32]. Finally, we focus on issues relating to parameter selection and running the model, as well as classifying StaMEs derived from both simulated and real data into a limited number of types (in our case 8 categories using hierarchical clustering methods). The real data relate to the movement data of barn owls (*Tyto alba*) in the Harod Valley, Israel, as more fully discussed in other contexts elsewhere [26, 33, 34].

# 2 Model Framework

## 2.1 Arrays and movement kernels

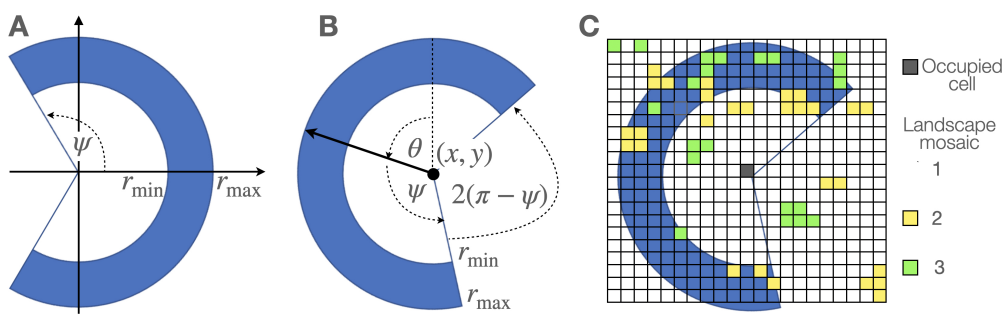


Figure 1: Kernels (**A.** area in blue) are specified in terms of the rims of sectors of circles with subtending angle $2\psi$ and inner and outer rim radii $r^{\min}$ and $r^{\max}$ respectively. Kernels become step-selection functions when located at a point $(x,y)$, provided an angle of heading $\theta$ (**B.**), placed on a landscape (**C.**) with cells of different levels of attraction or repulsion (green, yellow, and white squares) and admissibility (all cells that overlap with the kernel), and associated with a step selection rule $\mathscr{R}$.

In this paper, we focus on two dimensional models, which of course is only appropriate for some species (e.g., most terrestrial species) but not others (e.g., deep diving marine species or birds use thermals to gain height). For some birds, such as the barn owl data analyzed in Section 5.3 of this paper, accurate tracking in 3 dimension is infeasible and the height component of their flight is much smaller than the surface two dimensional components of their flight. Our model is thus implemented on a landscape represented by an $n^{\text{row}} \times n^{\text{col}}$ cellular array such that cell$(a,b)$ has value $c_{ab,t}$ at time $t$: i.e.,

$$\mathscr{A}(t) = \left\{ c_{ab,t} \middle| a = 1, \cdots, n^{\text{row}},\ b = 1, \cdots, n^{\text{col}} \right\} \tag{1}$$

Also, the topology of this array can be selected to be `topology = torus` (top-bottom and left-right continuity identification) or `plane` (top, bottom, left and right boundaries).

Each cell is identified both by its location $(a,b)$ in the array, and by Euclidean coordinates $(x_a, y_b)$ at the lower left corner of cell$(a,b)$. Depending on the units used to measure our Euclidean landscape, we define increments $\Delta x$ and $\Delta y$ such that $n^{\text{row}}\Delta x$ and $n^{\text{col}}\Delta y$ provide the desired dimensions for our landscape. Scaling is important when considering how far individuals are likely to move in one unit of time when in different movement modes and thus the scaling of

time is linked to the scaling of space in real applications. In theoretical studies not linked to empirical data, however, it will be convenient to set $\Delta x = \Delta y = 1$ and to set a parameter $\Omega_{\text{scale}} = 10$ to scale space with respect to time such that the greatest distance an individual is likely to travel in one time step is given by $\Omega_{\text{scale}}\Delta x$.

The model simulates movement of an individual over this landscape, with one step executed at each tick $t$ of the simulation clock for $t = 0, \cdots, n^{\text{time}}$. If an individual is at a point $(x_t, y_t)$ at time $t$ and moves to cell$(a,b)$, where its new location is now $(x_{\text{next}}, y_{\text{next}}) = (x_a, y_b)$, then the distance moved, denoted as $\rho_{ab}(x_t, y_t)$ is defined as

$$\rho_{ab}(x_t, y_t) = \sqrt{(x_a - x_t)^2 + (y_b - y_t)^2} \tag{2}$$

The angle of heading, denoted as $\theta_{ab}(x_t, y_t) \in [-\pi, \pi]$, as measured from the positive horizontal (i.e., the axis $x \geq 0$), is defined in terms of the so-called atan2 function as

$$\theta_{ab}(x_t, y_t) = \begin{cases} \arctan\left(\frac{y_b - y_t}{x_a - x_t}\right) & \text{if } x_a > x_t \\ \frac{\pi}{2} - \arctan\left(\frac{x_a - x_t}{y_b - y_t}\right) & \text{if } y_b > y_t \\ -\frac{\pi}{2} - \arctan\left(\frac{x_a - x_t}{y_b - y_t}\right) & \text{if } y_b < y_t \\ \arctan\left(\frac{y_b - y_t}{x_a - x_t}\right) \pm \pi & \text{if } x_a < x_t \\ \text{undefined} & \text{if } x_a = x_t \text{ and } y_b - y_t \end{cases} \tag{3}$$

Whenever the angle of heading is reported as ranging on $[0, 2\pi]$, it should be transformed to range over $[\pi, \pi]$ to ensure that computations of the turning angle and its absolute values are computed correctly (see Eq A.4 in the Appendix A.2).

Movement from current locations $(x_t, y_t)$ to a set of neighboring cells$(a,b)$ is computed in terms of kernels $K_\alpha$, $\alpha = 1, \cdots, n^{\text{stame}}$ belong to a set $\mathscr{K}$, where each kernel $K_\alpha$ is defined as the rim of the sector of a circle centered on the origin, with rim dimensions $r_\alpha^{\min}$ and $r_\alpha^{\max}$ and sector angle $2\psi_\alpha$ (Fig 1A) and includes an additional parameter $t_\alpha^{\text{s}}$ that influences the amount of time spent consecutively using kernel $K_\alpha$:

$$\mathscr{K} = \left\{ K_\alpha = K\left(r_\alpha^{\min}, r_\alpha^{\max}, \psi_\alpha\right) \middle| \alpha = 1, \cdots, n^{\text{stame}} \right\} \tag{4}$$

Of course, if $r_\alpha^{\min} = 0$, then the rim is actually a sector (slice) of a circle of radius $r^{\max}$.

The kernels $K_\alpha$ become step-selection functions $\tilde{K}_t^\alpha$ when anchored at time $t$ at a point $(x_t, y_t)$ with heading angle $\theta_t$ and 'time spent in current movement mode' variable $t_t^{\text{s}}$ (Fig 1B). They are associated with a set of movement rules $\mathscr{R}_{\lambda_\alpha}(\text{params})$, $\lambda_\alpha \in \{1, \cdots, n^{\text{rule}}\}$, where "params" are any parameters in the procedure that it is convenient to highlight. In particular, params includes a switching parameter $\hat{t}_\alpha^{\text{s}}$ that controls the probability of switching out of the current kernel as a function of $t_t^{\text{s}}$ and neighborhood cell value parameter $\hat{c}_\alpha^{\text{nbh}}$: i.e.,

$$\begin{aligned} \tilde{K}_t^\alpha\left(r_\alpha^{\min}, r_\alpha^{\max}, \psi_\alpha; x, y, \theta, \mathscr{R}_{\lambda_\alpha}(\hat{t}_\alpha^{\text{s}}, \hat{c}_\alpha^{\text{nbh}})\right) = K_\alpha \text{ at time } t \text{ anchored at } (x, y, \theta) \text{ and associated with step-selection} \\ \text{rule } \mathscr{R}_{\lambda_\alpha}(\hat{t}_\alpha^{\text{s}}, \hat{c}_\alpha^{\text{nbh}}) \text{ for some } \lambda_\alpha \in \{1, \cdots, n^{\text{rule}}\}, \quad \alpha = 1, \cdots, n^{\text{stame}} \end{aligned} \tag{5}$$

## 2.2 Landscape and individual dynamics

The initial set of landscape values $\mathscr{A}(0)$ (Eq 1) can either be read in or generated using an algorithm that, for example, either assigns cell values at random or generates patches of high valued cells in a matrix of low-value cells and even barrier cells in more complex versions of the model than presented here. We have implemented an algorithm for constructing either of these two cases using three parameters and default starting values of $c_{ab,0} = 0$ or 1: a parameter value $p^{\text{seed}}$ for laying down the first cells in patches at random, a parameter $p^{\text{cont}}$ for building patches into neighborhoods, and a parameter $n^{\text{cont}}$ for controlling the expected sizes of these patches. At the start of a simulation an individual animal has a state $h_0$, which is a representation of, for example, stored energy.

**Patchy landscape generation**

To initialize the landscape either read in an initial landscape file $\mathscr{A}(0)$ (Eq 1) or select one of the following two algorithms designated as $\texttt{RAM}_0^{\text{patch}}$ and $\texttt{RAM}_1^{\text{patch}}$, following procedures laid out in the Section 4 (RAM is an acronym for Runtime Alternative Module and will be described in more detail later).

Randomly located regular patches ($\texttt{RAM}_0^{\text{patch}}$)

- L.1 Default state: Set all cell values $c_{ab,0} = 0$ for $a = 1, \cdots, n^{\text{rows}}$, $b = 1, \cdots, n^{\text{cols}}$
- L.2 Lay down an initial set of patch seeds by switching the value of cell$(a,b)$ from $c_{ab,0} = 0$ to $c_{ab,0} = 1$ with probability $p^{\text{seed}}$ for $a = 1, \cdots, n^{\text{rows}}$, $b = 1, \cdots, n^{\text{cols}}$.
- L.3 For $a = 1, \cdots, n^{\text{rows}}$, $b = 1, \cdots, n^{\text{cols}}$ if $c_{ab,0} = 1$ then switch all cells that lie within a Moore neighborhood of radius $n^{\text{cont}}$ to cell$(a,b)$ to 1. This includes cells that may already have been switched to 1 because of their proximity to some other cell that has already been switched to 1.

Randomly located irregular patches ($\texttt{RAM}_1^{\text{patch}}$)

- L.4 Default state: Set all cell values $c_{ab,0} = 0$ for $a = 1, \cdots, n^{\text{rows}}$, $b = 1, \cdots, n^{\text{cols}}$
- L.5 First pass: Lay down an initial set of patch seeds by switching the value of cell$(a,b)$ from $c_{ab,0} = 0$ to $c_{ab,0} = 1$ with probability $p^{\text{seed}}$ for $a = 1, \cdots, n^{\text{rows}}$, $b = 1, \cdots, n^{\text{cols}}$.
- L.6 Second pass: For $a = 1, \cdots, n^{\text{rows}}$, $b = 1, \cdots, n^{\text{cols}}$ if $c_{ab,0} = 0$, and the sum of the values of cell$(a,b)$'s 4 neighbors (von Neumann neighborhood) is $\text{Nsum}_4(a,b)$ then switch cell$(a,b)$ from $c_{ab,0} = 0$ to $c_{ab,0} = 1$ with probability $1 - (1 - p^{\text{cont}})^{\text{Nsum}(a,b)}$
- L.7 Additional passes: Repeat step (c) $n^{\text{cont}}$ times (with the second pass corresponding to $n^{\text{cont}} = 1$).

The the default for the initial value $h_0$ of an individual is set 10. Other values can be entered as discussed in the simulation parameter setup below.

We note that if $p^{\text{cont}} = 0$, then cells in the array are randomly assigned a value 1 with probability $p^{\text{seed}}$

**Dynamic updating**

The cell array values $c_{ab,t}$ and individual value $h_t$ for $t = 0, \cdots, n^{\text{time}}$ are updated to account for the possibility that the individual gathers or extracts resources from cells as it moves over the landscape. This extraction may only take place during the implementation of some movement modes but not others.

Here we account for changes in these values as follows. The individual's value $h_t$ changes over time as it acquires resources when occupying cells of value $c_{ab,t} > 0$. It also incurs a cost $\kappa^{\text{sub}}$ per unit distance moved at each time step. Each time an individual occupies a cell$(a,b)$ it removes some resources $f^{\text{remove}}$. If we assume that removal is a resource density independent process, with $f^{\text{remove}} = \min\{\kappa^{\text{add}}, c_{ab,t}\}$ then the following updating rules for the value-state of individuals ($h$) and cells ($c$) and parameters $\kappa^{\text{add}} \in [0,1]$ and $\kappa^{\text{sub}}$ apply:

In moving from $(x_t, y_t)$ at time $t$ to cell$(a,b)$, we have the resource density-independent process

$$\begin{aligned} h_{\text{next}} &= \max\left\{0,\ h_t + \min\left\{\kappa^{\text{add}},\ c_{ab,t}\right\} - \kappa^{\text{sub}}\rho_{ab}(x_t, y_t)\right\} \\ c_{ab,\text{next}} &= c_{ab,t} - \min\left\{\kappa^{\text{add}},\ c_{ab,t}\right\} \end{aligned} \tag{6}$$

We can make this extraction dependent on the density of resource using the form of $f^{\text{remove}} = \frac{\kappa^{\text{add}} c_{ab,t}}{\kappa^{\text{add}} + c_{ab,t}}$, which is familiar to those who model consumer-resource interactions [35]. We can also allow for growth of the resource back to its carrying capacity of 1 at a a rate $\kappa^{\text{grw}}$ when completely removed (e.g., for grasses this represents regrowth from an intact rootstock, etc). Finally we can also make the cost of travel size (energy value) dependent on a suitable scaling

constant $\kappa^{\text{scl}} \geq 1$ and multiplying $\kappa^{\text{sub}}\rho_{ab}(x_t, y_t)$ by, for example, $\frac{1+h_t}{\kappa^{\text{scl}}+h_t}$. In this case, as $\lim_{h_t\to\infty} \frac{h_t+1}{h_t+\kappa^{\text{scl}}} \to 1$ and at $h_t = 0$ we obtain the factor $\frac{1}{\kappa^{\text{scl}}} < 1$. In this case, we obtain the equations:

In moving from $(x_t, y_t)$ at time $t$ to cell$(a,b)$, we have the resource density-dependent process

$$\begin{aligned} h_{\text{next}} &= \max\left\{0,\ h_t + \frac{\kappa^{\text{add}} c_{ab,t}}{\kappa^{\text{add}} + c_{ab,t}} - \left(\frac{h_t+1}{h_t+\kappa^{\text{scl}}}\right)\kappa^{\text{sub}}\rho_{ab}(x_t,y_t)\right\} \qquad (7) \\ c_{ab,\text{next}} &= \min\left\{1, c_{ab,t} + \kappa^{\text{grw}}\left(1 - c_{ab,t}\right)\right\} - \frac{\kappa^{\text{add}} c_{ab,t}}{\kappa^{\text{add}} + c_{ab,t}} \quad \text{for a currently occupied patch cell}(a,b) \\ c_{ab,\text{next}} &= \min\left\{1, c_{ab,t} + \kappa^{\text{grw}}\left(1 - c_{ab,t}\right)\right\} \quad \text{for all unoccupied patch cell}(a,b) \end{aligned}$$

We note that the simulation will stop either at $t = n^{\text{time}}$ or at $t^{\text{stop}}$ if $h(t^{\text{stop}}) = 0$.

In addition to the state value $h$ of the individual, we will also keep track of the time $t_t^{\text{s}}$ it has spent in its current movement mode. Thus, we will update $t_t^{\text{s}}$ as follows

$$t_{t+1}^{\text{s}} = \begin{cases} 0 & \text{every time the individual switches its movement mode} \\ t_t^{\text{s}} + 1 & \text{if the individual remains in the same movement mode} \end{cases} \qquad (8)$$

Thus at the start of the simulation, no matter the starting kernel, we initialize $t^{\text{s}}(0) = 0$.

## 2.3 Movement kernels

In our Numerus ANIMOVER_1 RAMP, we limit the implementation to at most two movement modes and hence two StaME kernels $K_\alpha$, $\alpha = \text{wp}, \text{bp}$. More specifically, we use $\alpha = \text{wp}$ to produce within-patch movement tracks and $\alpha = \text{bp}$ to produce between patch movement tracks. In addition, when $\alpha = \text{bp}$ we employ a vision kernel $K_{\text{vis}}(0, r_{wp}^{\max}, \pi/2)$ which allows an individual to see patches peripherally and in front of it to the maximum radius of its next $\alpha = \text{bp}$ movement-mode step.

### Kernel parameters

In general we expect within patch steps to be smaller than between patch steps, and within patch turning to be larger than between patch turning, although we have the flexibility to set up contrary scenarios. For each of the two movement modes at the start of a simulation, values for the triplets $\left(r_{\text{wp}}^{\min}, r_{\text{wp}}^{\max}, \psi_{\text{bp}}\right)$ and $\left(r_{\text{bp}}^{\min}, r_{\text{bp}}^{\max}, \psi_{\text{bp}}\right)$ are set up. If our expectation is followed, then we $r_{\text{wp}}^{\max} < r_{\text{bp}}^{\max}$ and $\psi_{\text{bp}} < \psi_{\text{wp}}$ will hold. We may also expect $r_{\text{wp}}^{\min} = 0$ if the individual may choose to remain in the same cell for more than 1 time period and $r_{\text{bp}}^{\max} = r^{\min} + 1\text{or}2$ if within patch movements are considerably less than between patch movement where we suggest setting $r_{\text{bp}}^{\max} = 10$. We again stress that the user has the flexibility to set and scale this values arbitrarily.

All that remains in implementing the algorithm depicted in Fig 2, is specification of the parameters needed to implement our step-selection functions (SSFs, [36, 37, 38]) $\tilde{K}^{\text{wp}}$ and $\tilde{K}^{\text{wp}}$ using the rules $\mathscr{R}_{\lambda_{\text{wp}}}(\hat{t}_{\text{wp}}^{\text{s}}, \hat{c}_{\text{wp}}^{\text{nbh}})$ and $\mathscr{R}_{\lambda_{\text{bp}}}(\hat{t}_{\text{wp}}^{\text{s}}, \hat{c}_{\text{wp}}^{\text{nbh}})$ that are specified next (the last of these parameter arguments applies to $\texttt{RAM}_1^{\text{update}}$, but not $\texttt{RAM}_0^{\text{update}}$). We note that these procedures are coded as the default RAMs for our Numerus ANIMOVER_1 RAMP. Other procedures, possibly using integrated step selection analysis (iSSA, [39]) or methods that include direction-biasing external points of attraction or repulsion [40], may be included by the user.

### Step-selection cells and probabilities

The step-selection rules $\mathscr{R}_{\lambda_\alpha}$ are procedures for updating the next location, angle of heading, and time spent in the current movement mode $(x_{\text{next}}, y_{\text{next}}, \theta_{\text{next}}, t_{\text{next}}^{\text{s}})$, as well as the kernel $\alpha_{\text{next}}$ to be used next. This is done in terms of the individuals current location, angle of heading and spent time $(x_t, y_t, \theta_t, t_t^{\text{s}})$, and its current movement mode, as driven by the kernel implementation $\tilde{K}_t^{\alpha}$. In our case we have two sets of rules, one that specifies within patch movement $(\mathscr{R}_{\text{wp}}(\hat{t}_{\text{wp}}^{\text{s}}, \hat{c}_{\text{wp}}^{\text{nbh}}))$ and one that specifies between patch movement $(\mathscr{R}_{\text{wp}}(\hat{t}_{\text{wp}}^{\text{s}}, \hat{c}_{\text{wp}}^{\text{nbh}}))$. These rules include situations where the topology of the landscapes is a bounded rectangle and normal search fails to find a next location.

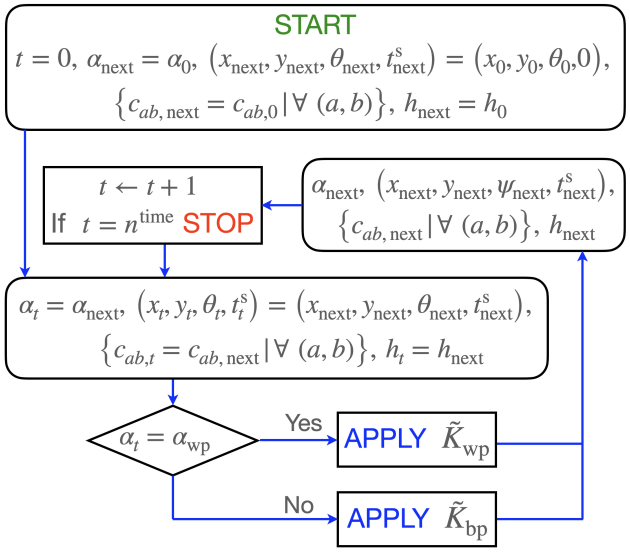


Figure 2: Algorithm used to simulate movement after the structural parameters $n^{\text{time}}$ (index $t$), $n^{\text{row}}$ (index $y$) and $n^{\text{col}}$ (index $x$) been entered, along with the topology of the landscape, the initial landscape values $\left\{c_{ab,0}|\forall(a,b)\right\}$, the initial state value $h_0$ of the individual, and the parameter triplets $\left(r^{\text{min}}_{\text{wp}},r^{\text{max}}_{\text{wp}},\psi_{\text{wp}}\right)$ and $\left(r^{\text{min}}_{\text{bp}},r^{\text{max}}_{\text{bp}},\psi_{\text{bp}}\right)$ that define the two kernels $K_{\text{wp}}$ and $K_{\text{bp}}$ respectively. Implementations $\tilde{K}^{\text{wp}}$ and $\tilde{K}^{\text{bp}}$ require the specification of step-selection procedures $\mathscr{R}_{\text{wp}}(\hat{t}^{\text{s}}_{\text{wp}},\hat{c}^{\text{nbh}}_{\text{wp}})$ and $\mathscr{R}_{\text{bp}}(\hat{t}^{\text{s}}_{\text{bp}},\hat{c}^{\text{nbh}}_{\text{bp}})$ respectively. The latter employs an additional kernel $K_{\text{vis}}$ that controls how the individual scans the landscape when it is moving between patches. The START, STOP, and APPLY commands, along with the flow arrows are show in different colors for additional clarity. The details of step selection procedures $\mathscr{R}_{\text{wp}}(\hat{t}^{\text{s}}_{\text{wp}},\hat{c}^{\text{nbh}}_{\text{wp}})$ and $\mathscr{R}_{\text{bp}}(\hat{t}^{\text{s}}_{\text{bp}},\hat{c}^{\text{nbh}}_{\text{bp}})$ can be found in Appendix A.3.

In computing our step-selection procedures we will make use of the set $\mathscr{C}^{\alpha}_t(x_t,y_t,\theta_t)$ defined as follows:

$$\mathscr{C}^{\alpha}_t(x_t,y_t,\theta_t)=\left\{c_{ab,t}\middle|\text{cell}(a,b)\text{ overlaps with }\tilde{K}^{\alpha}_t\right\},\quad \alpha=\text{wp or bp} \tag{9}$$

We will also make use of the following sets of probabilities

$$\mathscr{P}^{\alpha}_t(x_t,y_t,\theta_t)=\left\{p_{ab}=\frac{c_{ab}}{\sum_{\text{cell}(a,b)\in\mathscr{C}^{\alpha}_t}}\middle|\forall\ \text{cell}(a,b)\in\mathscr{C}^{\alpha}_t\right\}\quad \alpha=\text{wp or bp} \tag{10}$$

Finally, we will make use of the probabilities $p_{\alpha}(t^{\text{s}})$ of continuing to use StaME $K_{\alpha}$ when having used this StaME for the past $t^{\text{s}}$ time steps

$$p_{\alpha}(t^{\text{s}})=\frac{e^{-4\left(t^{\text{s}}-\hat{t}^{\text{s}}_{\alpha}\right)}}{1+e^{-4\left(t^{\text{s}}-\hat{t}^{\text{s}}_{\alpha}\right)}}\quad \alpha=\text{wp or bp} \tag{11}$$

An outline of the implementation of step selection procedures $\mathscr{R}_{\text{wp}}(\hat{t}^{\text{s}}_{\text{wp}},\hat{c}^{\text{nbh}}_{\text{wp}})$ and $\mathscr{R}_{\text{bp}}(\hat{t}^{\text{s}}_{\text{bp}},\hat{c}^{\text{nbh}}_{\text{bp}})$ is provided in Fig 2 with details provided in Appendix A.3.

# 3 Movement Paths and StaME Extraction

A movement track, whether simulated or empirical, in the first instance has a representation as a relocation time series of $n^{\text{time}}$ points—i.e., a walk

$$W=\left\{\left(t;x(t),y(t)\right)\middle|t=0,\ldots,n^{\text{time}}\right\} \tag{12}$$

Such tracks can be generated using simulations models, as we discus in some depth in the next section. In the case of empirical data, though, data preprocessing and filtering [41] are needed to get rid of spurious or problematic points

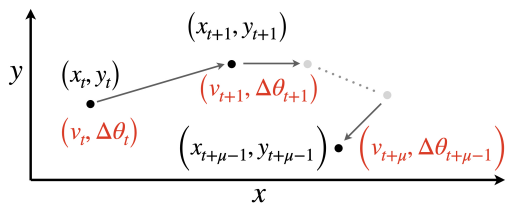


Figure 3: A relocation time series $W$ (Eq 12) is plotted in the $xy$-plan with computed velocities $v_t$ (Eq A.2) and turning angles $\Delta\theta_t$ in red (Eq A.4) for a segment of $\mu$ points at times $\tau = t, t+1, \cdots, t+\mu-1$. These values, or others (e.g., $v_t^{\text{per}}$ and $v_t^{\text{tan}}$; see Eq A.5 and [24]) can then be used to compute statistical elements $\text{Seg}_z$ (Eq 13) from a series of consecutive elements containing $\mu$ relocation points as, for example, in Eq 13

or fill in missing points. From walk Eq 12, both velocity and turning-angle time series can be generated, as outlined in Appendix A.2 and elsewhere [15]. Various other statistics (e.g., travel distance, net displacement) can be extracted as well and used as variables to define a set of basic statistical movement elements (StaMEs), following methods described next.

## 3.1 Creating StaMEs

The method described here to create StaMEs uses the velocity ($V$) and turning-angle ($\Delta\Theta$) time series derived from walk $W$, as described in Eq A.1-A.4 (Appendix A.2). All points in our empirical time series that produced unrealistically large velocities were then removed. Although the putative maximum sustained flying speeds of barn owls have been observed in the range of 6-8 m/s (17.9 mph) [42, 43], to be conservative, we only removed points that represented unrealistically speeds. In our case, this amounted to a handful of points with velocities in excess of 75 m/s (168 mph); and we note that the average speed of the fastest 1 minute segments in our analysis below (Table 2) turns out to be a credible 3-4 m/s.

In the case of our simulated data, we normalized the entries of our cleaned $V$ and $\Delta\Theta$ time series by dividing each of the entries $v_t$ in $V$ by $v_{\max} = \max_{t=1,\cdots,n^{\text{time}}}\{v_t \in V\}$ ($= 10$ in our simulations) and dividing each of the entries in $\Delta\Theta$ by $2\pi$ to obtain variables on the ranges $[0,1]$ and $[-1/2, 1/2]$ respectively. In the case of our empirical data, we only normalized the turning angles since the velocities had the physical units m/s and we wanted to reflect this in our results.

Next, we parsed the normalized velocity and turning-angle time series into $z = 1, \cdots, n^{\text{seg}}$ segments each of length $\mu$ (Fig 3) with regard to $t$: in segments with missing points (these points may be filled in using an appropriate interpolation method) we kept the same length of segment and just adjusted for the reduced number of points (the errors from such missing points are likely to be inconsequential when the number missing is a few percent or less). The total number of segments so obtained was $n^{\text{seg}} = \lfloor \frac{n^{\text{time}}-1}{\mu} \rfloor$, with some points left over when $n^{\text{time}} - 1$ was not exactly divisible by $\mu$.

We then calculated a set of statistics related to the $\mu$ normalized (in the simulated data only) velocities (equivalent to step lengths) and turning angles for each of the segments $z$ in our time series data. Although various sets of statistics can be used (such as persistent and tangential velocities [24]), we settled upon mean velocities $V_z$ and mean absolute turning angles $|\Delta\Theta|_z$ for each segment and associated standard deviations $\text{SD}_z^V$ and $\text{SD}_z^{|\Delta\Theta|}$. Also to pick up any possible circular motion type biases in movement, we computed a normalized net displacement ($\Delta^\rho$) statistic for each segment (i.e., the distance between the first and last points of each segment divided by quantity equal to the the number of points multiplied by the mean step-length). Specifically, for velocity and turning-angle means and standard deviations (SD) (normalized where appropriate), as well as net displacements, we defined a set of segments $\mathscr{S}_\mu$ (note

below that $v_{\max} = 1$ for the empirical data and $\mu$ is adjusted for segments with missing points)

$$\mathscr{S}_\mu = \left\{ \mathrm{Seg}_z\big(V_z, \mathrm{SD}_z^V, |\Delta\Theta|_z, \mathrm{SD}_z^{|\Delta\Theta|}, \Delta_z^\rho\big) \,\big|\, z = \lfloor \frac{t}{\mu} \rfloor + 1,\ t = 0, \cdots, n^{\mathrm{seg}} - 1 \right\} \text{ such that}$$
$$V_z = \frac{\sum_{\tau=t}^{t+\mu-1} v_\tau}{v_{\max}\mu} \text{ with st. dev. } \mathrm{SD}_z^V,\ |\Delta\Theta|_z = \frac{\sum_{\tau=t}^{t+\mu-1} |\Delta\theta_\tau|}{2\pi\mu} \text{ with st. dev. } \mathrm{SD}_z^{|\Delta\Theta|} \tag{13}$$
$$\text{and } \Delta_z^\rho = \frac{\sqrt{(x_{t+\mu} - x_t)^2 + (y_{t+\mu} - y_t)^2}}{\mu V_z}$$

## 3.2 Mapping StaME Centroids to Kernels

We first applied our segmentation procedure Eq 13 to data obtained from simulating the movement of an individual using a single kernel $K_\alpha = K(r_\alpha^{\min}, r_\alpha^{\max}, \psi_\alpha)$ on an unstructured (static, homogeneous) landscape. If one generates a segmentation set $\mathscr{S}_\mu$ (Eq 13) from these simulation data, then using a variety of kernels with different admissible $r_\alpha^{\min}$, $r_\alpha^{\max}$ and $\psi_\alpha$ values, one can build up a discrete map represented by the function

$$\mathscr{F}_\mu^{\mathrm{hom}} :\ \left(r_\alpha^{\min}, r_\alpha^{\max}, \psi_\alpha,\right) \mapsto \left(\overline{V}_\alpha, \overline{\mathrm{SD}^V}_\alpha, \overline{|\Delta\Theta|}_\alpha, \overline{\mathrm{SD}^{|\Delta\Theta|}}_\alpha, \overline{\Delta^\rho}_\alpha\right) \quad \text{for all possible } K_\alpha \tag{14}$$

We note that we have not subscripted the quantities in the image of this mapping with the parameter $\mu$. The reason is that the difference in values obtained for different values of $\mu$ are just variances associated with sampling and therefore are not consequential. However, we maintain the subscript $\mu$ on the mapping itself to remind ourselves that a value needs to be selected before this set can be generated. We also note, in the case of an unbiased walk, the statistic $\overline{\Delta^\rho}$ may be ignored, because the realized movement has no prevailing circular bias to its motion. Under these circumstances, $\overline{\Delta^\rho}$ is perfectly correlated with the other statistics defining each segment because clockwise and counterclockwise movements are equally likely. Additionally, values obtained should be relatively insensitive to $\mu$ when $\mu$ and the length $n^{\mathrm{time}}$ of the track itself are sufficiently large to ensure that the law of large numbers is at play.

We then applied this segmentation procedure to data obtained from an ANIMOVER_1 two-kernel patch simulations ($K_{\mathrm{wp}}$ and $K_{\mathrm{pb}}$). Once the segmentation set $\mathscr{S}_\mu$ had been generated (Eq 13), we carried out a hierarchical cluster analysis using Ward's method (Appendix A.4) to obtain a set of $k$ centroids represented by the set

$$\mathscr{X}_{0,\mu,k}^{\mathrm{patch}} = \left\{ \left\{ \left(\overline{V}_\iota, \overline{\mathrm{SD}^V}_\iota, \overline{|\Delta\Theta|}_\iota, \overline{\mathrm{SD}^{|\Delta\Theta|}}_\iota, \overline{\Delta^\rho}_\iota\right) | \iota = 1, \cdots, k \right\}_{K_{\mathrm{wp}}, K_{\mathrm{bp}}, \mathscr{L}_0^{\mathrm{patch}}} \middle| \forall \text{ relevant combinations of } K_{\mathrm{wp}}, K_{\mathrm{bp}}, \mathscr{L}_0^{\mathrm{patch}} \right\} \tag{15}$$

As in the case of quantifying selected points of the discrete map $\mathscr{F}_\mu^{\mathrm{hom}}$, so can we quantify selected points of a discrete map $\mathscr{F}_{0,\mu,k}^{\mathrm{patch}}$ by computing $k$ centroids in the set $\mathscr{X}_{0,\mu,k}^{\mathrm{patch}}$ from simulations of ANIMOVER_1 using different combinations of $K_{\mathrm{wp}}$ and $K_{\mathrm{bp}}$ kernels (i.e., step-selection procedures $\mathscr{R}_{\mathrm{wp}}$ and $\mathscr{R}_{\mathrm{bp}}$) with individuals moving over generated landscapes $\mathscr{L}_0^{\mathrm{patch}}$. This mapping, in terms of the parameters used to generate it, can be expressed by:

$$\mathscr{F}_{0,\mu,k}^{\mathrm{patch}} : \left(K_{\mathrm{wp}}, K_{\mathrm{bp}}, \mathscr{L}_0^{\mathrm{patch}}\right) \mapsto \mathscr{X}_{0,\mu,k}^{\mathrm{patch}} \tag{16}$$

In our computations, we selected our cluster number $k$ to be 8 rather than 2, (see Appendix A.4 for details) even though we only had two movement modes. The reason for this is that some segments in our segmentation process will be mixtures of the two modes rather than homogeneous strings of points generated by either one or other of the two modes (see Fig C.1) while we expected only 2 of the 8 clusters to contain relatively homogeneous movement mode segments. Of course, the extent to which mixed versus pure movement mode segments arises depends both on the length of segments and the frequency at which the two movement modes switch between each another. The shorter the segments, or the less frequent the switching, the more likely any segment represent a series of locations generated by a single movement mode.

We can use our simulation model to numerically construct a mapping $\mathscr{F}_{0,\mu,k}^{\mathrm{patch}}$ of a set $\left(r_\alpha^{\min}, r_\alpha^{\max}, \psi_\alpha\right)$ of kernel arguments onto a set of selected cluster centroid statistics $\left(\overline{V}_{\iota_\alpha}, \overline{\mathrm{SD}^V}_{\iota_\alpha}, \overline{|\Delta\Theta|}_{\iota_\alpha}, \overline{\mathrm{SD}^{|\Delta\Theta|}}_{\iota_\alpha}, \overline{\Delta^\rho}_{\iota_\alpha}\right)$, as outlined in Appendix B. By way of illustration, we generate one point of the map $\mathscr{F}_\mu^{\mathrm{hom}}$ in this paper. The range of values that may be

useful to generate in a multi-point construction of this map will depend on the kind of empirical to which this mapping is fitted. Such an exercise is, thus, best left to a detailed study that explores the structure of given set of empirical movement data, such as an extended set of the owl data discussed below—a set containing data collected from at least several tens of individuals (e.g., as in [26, 33]).

# 4 Numerus RAMPs

## 4.1 RAMP Construction

The java-based Numerus Model Builder Designer (**NMB Designer**) Platform was used to code the model and then generate a **RAMP** (runtime alterable model platform) our **Numerus ANIMOVER_1** Application (with future versions 2, 3, etc. planned to include multidimensional landscapes, additional movement modes, and interacting individuals). The application takes the form of a portable file that, as described elsewhere [44, 45], and is played on the free downloadable NMB Studio application.

The flexibility of NMB RAMPs, beyond manipulating parameter values using its sliders (Fig 4E and F) and windows (Fig 4E and F), is facilitated by its runtime alterable modules (`RAM`s, Fig 4G, Fig 5). These `RAM`s provide the user with the ability to chose alternative formulations of component parts of the model just priori to rerunning a current simulation or recode those component parts with alternate expressions. Thus, for example, in updating the value equations $h_t$ and $c_{ab,t}$, the RAMP uses the default $\texttt{RAM}_0^{\text{update}}$, which codes Eq 6, as a default procedure, or the user can select $\texttt{RAM}_1^{\text{update}}$, which codes Eq 7, as an alternate procedure (Fig. 5). In addition, the user can create a second alternative `RAM` by opening a new `RAM` window and inserting and saving code for customized equations, although parameters beyond the three already available as sliders at the console will have to be given fixed values (i.e., no new sliders can be created or added outside of upgrading the RAMP using the Numerus Designer Platform).

## 4.2 RAMs

Two or more versions of the `RAM`s listed below exist: first a default version ($\texttt{RAM}_0$), a selectable alternative version ($\texttt{RAM}_1$), and perhaps additionally higher numbered versions or extemporaneously created versions, versions that can be added just prior to initiating a run of the model, and saved for future use. Thus, for example, Eq 6 for the value dynamics of individuals and cells are coded up as default $\texttt{RAM}_0^{\text{update}}$, while the more elaborate density-dependent version of these equations (Eq 7) are coded up as $\texttt{RAM}_1^{\text{update}}$.

ANIMOVER_1 includes the following `RAM`s:

**Initial landscape `RAM`s**

$\texttt{RAM}_0^{\text{patch}}$ (default) This `RAM` lays down a patchy landscape using rules L.1-L.3 to create irregularly-shaped (stochastic) patches stochastically placed at a specified density. It requires parameter values $p^{\text{seed}}$ to define the patch density, and $n^{\text{cont}}$ and $p^{\text{cont}}$ to generate the patch size and level of irregularity. Patches will be larger and more squarish when Moore is selected over von Neumann neighborhood.

$\texttt{RAM}_1^{\text{patch}}$ (alt 1) This `RAM` lays down a patchy landscape using rules L.4-L.7 to create regularly shaped patches (cubes using Moore neighborhood, diamonds using von Neumann neighborhood) stochastically placed at a specified density. It requires parameter values $p^{\text{seed}}$ to define the patch density, and $n^{\text{cont}}$ to generate patch size.

$\texttt{RAM}_{2\ldots}^{\text{patch}}$ Available to the user for coding customized methods for creating initial landscape structures.

**Resource Extraction `RAM`s**

$\texttt{RAM}_0^{\text{val}}$ (default): This `RAM` updates the values $h_t$ and $c_{ab,t}$ using Eg 6 and requires parameter values $\kappa^{\text{add}}$ and $\kappa^{\text{sub}}$.

$\texttt{RAM}_1^{\text{val}}$ (first alternative): This `RAM` updates the values $h_t$ and $c_{ab,t}$ using Eg 7 and requires parameter values $\kappa^{\text{add}}$, $\kappa^{\text{sub}}$ and $\kappa^{\text{grw}}$.

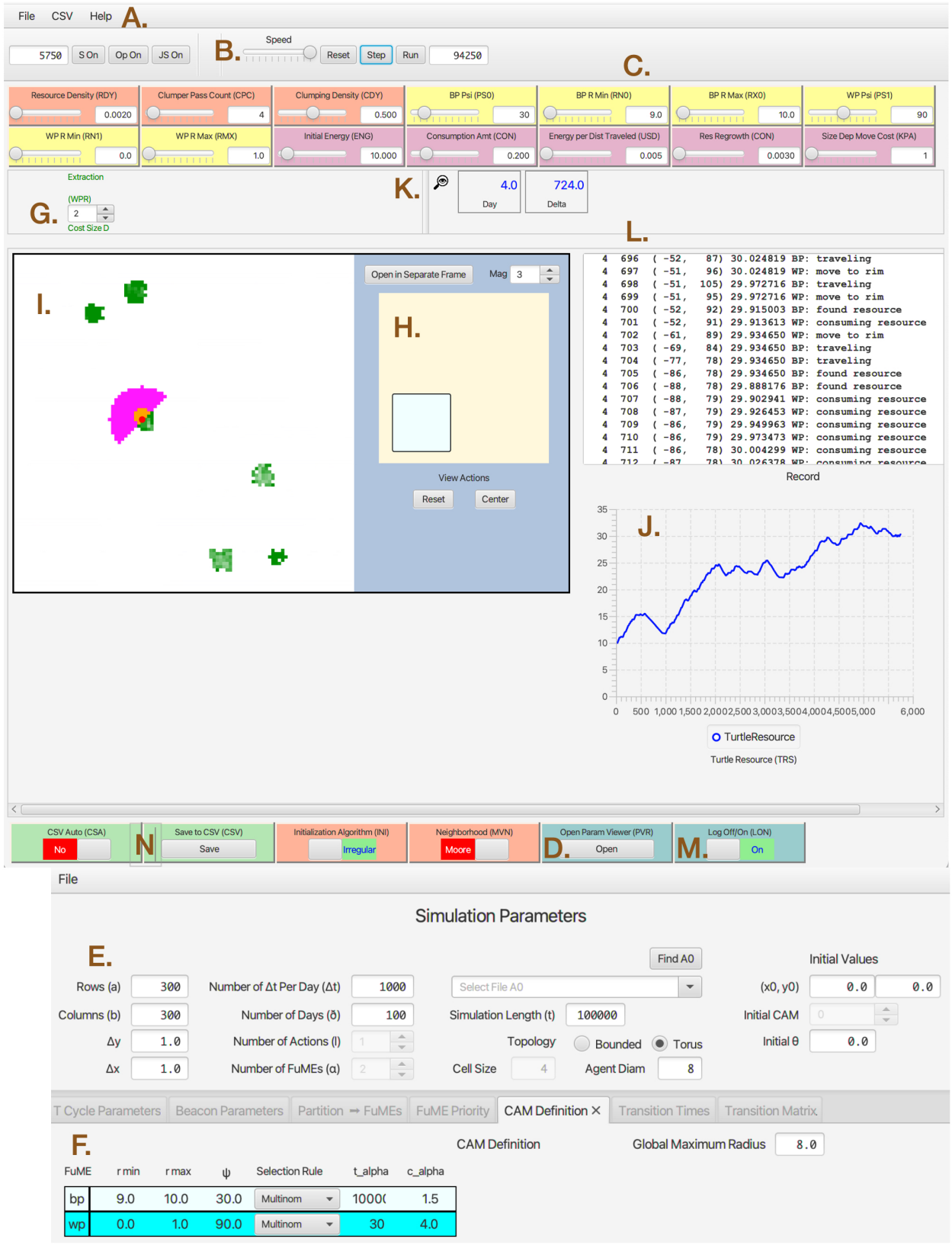


Figure 4: Caption on next page.

Figure 4: ANIMOVER_1 console as viewed in the Numerus Studio application. Labels A.-N. in brown are added for purposes of exposition. **A:** Pull down menus to load files. **B:** Run time controls allow for a reset in the middle or end of the run, step by step view of the changes (panels I and J), or automated run for full simulation time using parameters in panel E. **C:** Color coded sliders used to change parameter values at the start of or during a run (orange: landscape initialization; yellow: kernel parameters; lilac: consumer resource interactions; green: saving output; turquoise: parameter and monitoring windows. **D:** Button to open the parameter windows E & F. **E:** A form where various parameters that control the size and scaling of the run are set. **F:** A set of forms to enter between (bp) and within (wp) patch kernel parameters (angles in degrees). **G:** The wheel is used to select the $\texttt{RAM}^{\text{update}}$ to be used. **H:** A viewer of the whole landscape with inset that can be observed as magnified in viewer I. **I:** A mouse-manipulable viewer of the landscape (panel H) that can be moved (using right click and move) and zoomed. Shown in I. are the current location (red dot), five patches (green pixels, lighter one previously exploited), and the current kernel (orange and lilac pixels). **J:** A plot of the value $h_t$ of the agent over time. **K:** Record of current day and within day iteration with additional **L:** activity log and **M:** log switch. **N:** Switches and buttons to save end-of-run output.

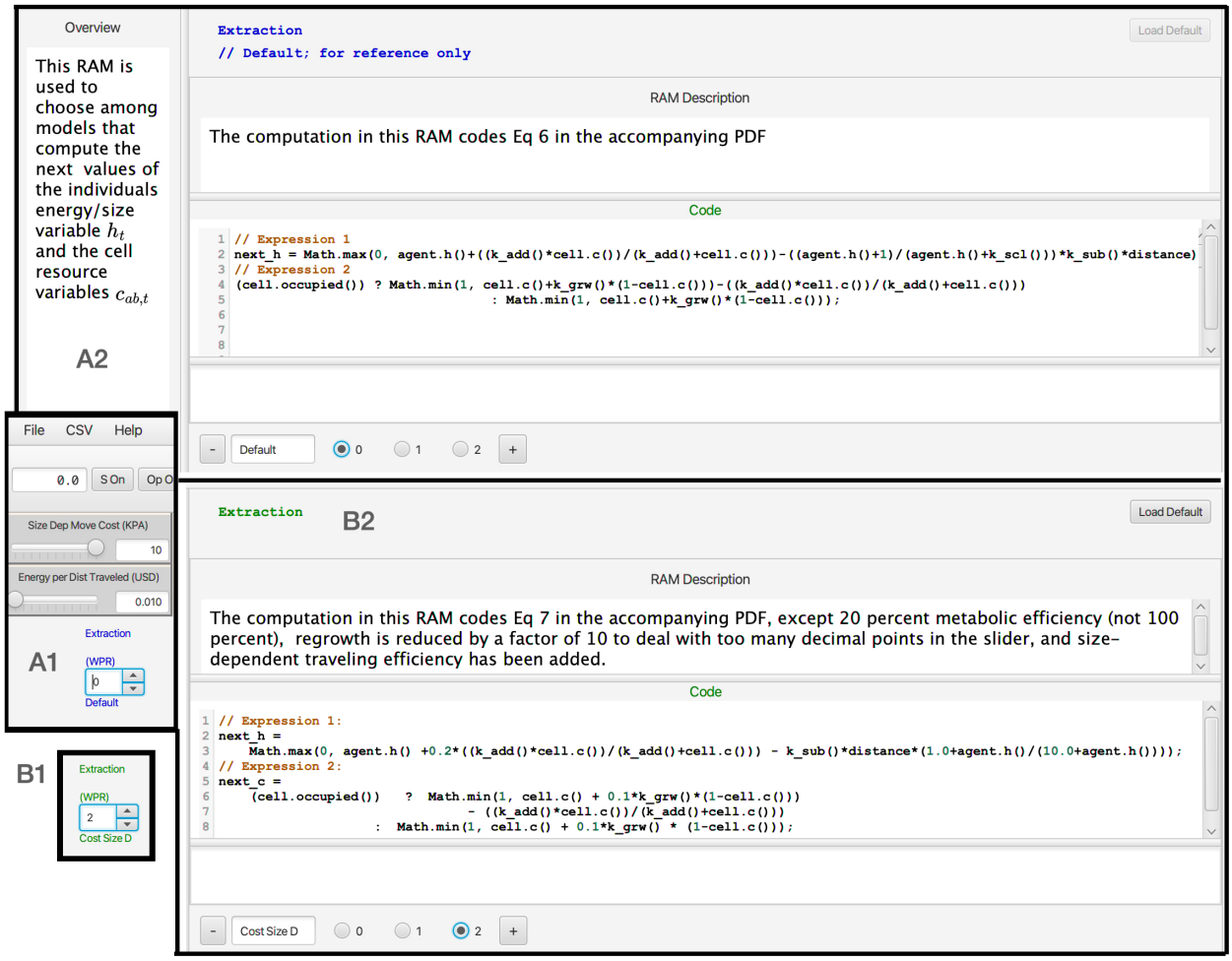


Figure 5: **A1:** The location on the ANIMOVER_1 console (**G.** in Fig 4) where the particular RAM to be implement in the current run is selected using the numbered roller. **A2:** The windows of the Default RAM showing the code that implements Eq 6. **B1:** This shows the roller selected to position 2 to implement the Extraction RAM expressed in Eq 7. Position 1 refers to a form of the equations not discussed in the text. **B2:** The windows of the Extraction RAM showing the code that implements Eq 7. The "+" sign at the bottoms of windows A2 and B2 allow the user to open a window for the user to supply their own customized code for updating the state of the individual $h_t$ and cells array values $c_{ab,t}$.

## 4.3 Parameter setup

The following set of parameters are needed to simulate the model. We also include some information on how our application console may be used to input some of the parameter values. Once this is done for those parameters that are either read in as a `FILE` (Fig 4A), entered prior to the simulation using a `FORM` (Fig 4E and F), specified using a `PULLDOWN` menu, or ignored—in which case `DEFAULT` vaules will be used, the Numerus ANIMOVER_1 will create a parameter values data file that can be download, edited and re-uploaded as needed. Other parameters will be entered and flexibly changed using a `SLIDER` (Fig 4C). Some of the these sliders will reflect values that are entered using a `FORM` while others will not be reflected in the form, but saved internally when the current simulation job is saved by the Numerus Studio Application Platform.

P.1 Size and scope parameters (`FORMS`). These are: $n^{\text{row}}$ (index $a$), $n^{\text{col}}$ (index $b$), $n^{\text{time}}$ (index $t$) and `topology = torus` or `plane` (`PULLDOWN`). We note that we have fixed $n^{\text{stame}} = 2$ (index $\alpha$), Entry of these numbers and topology type respectively specify the dimensions of the cellular array, the length of the simulation, and whether the simulation takes place on a torus or a bounded plane.

P.2 Scaling parameters $\Delta x$, $\Delta y$ (`FORMS` or `DEFAULT`). Entry of these two Cartesian scaling values are used to assign the $x$-$y$ coordinate values $(x_a, y_b)$ to each of the cells ($a = 1, \cdots, n^{\text{row}}$, $b = 1, \cdots, n^{\text{col}}$) in the simulation space $\mathscr{A}$. The default values for this are $\Delta x = \Delta y = 1$.

P.3 Cellular array $\mathscr{A}(0)$ (`FILE`) or generating parameters $p^{\text{seed}}$, $p^{\text{cont}}$ and $n^{\text{cont}}$ (`SLIDERS`). This is an appropriately configured data file (e.g., csv text) that will up uploaded to the application to provide the initial state of all the cells in the simulation space or the initial landscape will be generated using the specified parameters algorithm L.2.

P.4 Kernel definition parameters $K_\alpha$, $\alpha = \text{wp and bp}$ (`FORM` and `SLIDERS`). Enter the three arguments for each of two kernels: i.e., $\left(r_\alpha^{\min}, r_\alpha^{\max}, \psi_\alpha\right)$, $\alpha = \text{wp and bp}$, respectively.

P.5 Kernel implementation parameters for $\tilde{K}^\alpha$ (`SLIDER`). Enter the arguments $\left(\hat{t}_\alpha^{\text{s}}, \hat{c}_\alpha^{\text{nbd}}\right)$ for the step-selection procedures $\mathscr{R}_\alpha$, $\alpha = \text{wp and bp}$.

P.6 Initial values (various). The initial time is automatically taken to be $t = 0$. The initial location and angle is computed from the selection of values $(a, b)$ entered (`FORM`) or ignored (`DEFAULT`) and an initial value for the heading direction $\theta_0$ is entered (`FORM`) or ignore `DEFAULT`). The default starting cell is $a = \lfloor n^{\text{row}}/2 \rfloor$, $b = \lfloor n^{\text{col}}/2 \rfloor$, and default angle of heading is $\theta_0 = 0$. The actual starting location is thus $(x_a, y_b)$. The initial value $h_0$ for the individual must be entered (`Form`) or will be 10 (`DEFAULT`).

P.7 Initial kernel wp or bp (`PULLDOWN`). This will set the initial condition $\alpha_0$

P.8 Update parameters $\kappa^{\text{add}}$, $\kappa^{\text{sub}}$, $\kappa^{\text{grw}}$ and $\kappa^{\text{scl}}$ (`SLIDERS`). The first two of these parameters are used in Eq 6 or all four in Eq 7 for updating $h_t$ and $c_{ab,t}$.

P.9 Step selection rule parameters $\hat{t}_{\text{wp}}^{\text{s}}$, $\hat{t}_{\text{wp}}^{\text{s}}$, $\hat{c}_{ab,\text{wp}}$, $\hat{c}_{ab,\text{bp}}$ (`SLIDERS`). The first two will be integer value sliders between $[1, 1000]$, while the second two will be values to 1 dp between [0,5].

## 4.4 Simulation setups and units

If landscape pixels are $\Delta x = \Delta y = 10$ m and $n^{\text{row}} = n^{\text{col}} = 500$ then our 250,000 pixel landscape is $5^2 = 25$ km$^2$. If the units of $t$ are 1 m intervals, then in a day an individual will move up to $1440 = 60 \times 24$ times. If an individual walks at a persistent speed of 5 km per hour, which is $\frac{5000}{3600} = 1.39$ m/s, then an individual can cross either the length or breadth of our landscape in 1 hour, and a single pixel in 14 secs. In this case, the a maximum radius of $r^{\max} = 4$ would suffice, although a movement kernel covering short sprints would have a maximum radius of 4 or 6 times this value. For theoretical studies, boundary effects can be avoided during simulations by setting `topology=torus` (which makes the left and right columns of cells in the landscape array neighboring columns, and the top and bottom rows of cells neighboring rows).

### 4.5 Model output

During the course of a simulation, one can visual observed the run as it progresses in window I (Fig 4) of the ANIMOVER_1 console. One can also observe the sequence of movements for the current day stage of the simulation in window L if switch M is in the "On" position (Fig 4). At the end of the run a CSV file is automatically saved if switch N is in the "Yes" position or if the "Save to CSV" button N is pressed after the run is complete. The saved CSV file is headed by a list of all the parameter settings for the run. It also has the following columns of data consecutive generated at step of the model simulation: day, within-day step, $x$-location, $y$-location, distance moved angle of head (degrees), agent-state (resources), movement mode (bp or wp) (Fig C.1). An example of this output can be found in supplementary online file "TwoModeSimulationData" with a graphically depicted subset explained in Appendix C.1 (Output Data). In addition, links to downloading ANIMOVER_1 and to a RAMP Users Guide can be found in Appendix C.2.

## 5 Illustrative Examples

### 5.1 One point in the mapping of $\mathscr{F}_{\mu}^{\text{hom}}$

By way of illustration, we used ANIMOVER_1 to construct 1 point in the mapping $\mathscr{F}_{\mu}^{\text{hom}}(K)$ (Eq 14). Specifically, we used the kernel $K(0,6,\pi/2)$ to simulate the movement of an individual for 1500 steps over a homogeneous landscape. Using a $\mu = 15$ points segmentation interval we generated the set $\mathscr{S}_{15}$ (Eq 13) and then calculated the means of the statistical variables of the interest to obtain $\left(\overline{V}, \overline{\text{SD}^V}, \overline{|\Delta\Theta|}, \overline{\text{SD}^{|\Delta\Theta|}}\right) = (3.38, 1.23, 0.27, 27.6)$. Thus we identified one point in the mapping $\mathscr{F}_{\mu}^{\text{hom}}$ (Eq 14): specifically,

$$\mathscr{F}_{15}^{\text{hom}}: \ (0,5,\pi/2) \mapsto (3.38, 1.23, 0.27, 0.15) \tag{17}$$

where we reiterate that we did not compute the displacement variable because it is non-informative due to the lack of a circular bias to the motion. Other points in the mapping can of course be constructed, as outlined in M.1-4 above. The range of parameters use to compute the structure of $\mathscr{F}_{\mu}^{\text{hom}}$ depends on the range of the image space $\left(\overline{V}, \overline{\text{SD}^V}, \overline{|\Delta\Theta|}, \overline{\text{SD}^{|\Delta\Theta|}}\right)$ that needs to be covered. Interpolation can also be used, where desired to estimate points that are contained within the nodes of the lattice structure used to compute the mapping at discrete points in the range space (as discussed in Appendix B).

### 5.2 A two-movement mode simulation

We carried out a two-movement mode simulation on a patchy landscape that we manually stopped after 69,525 model steps. The parameters that we used for this simulation were:

#### Two movement mode simulation

For this simulation, we used the following parameters:

*Size and scaling.* $n^{\text{row}} = 300$, $n^{\text{col}} = 300$, $n^{\text{time}} = 100,000$ (as an upper limit) and `topology` = `torus`, $\Delta x = \Delta y = 1$

*Landscape generator.* Initialization algorithm = Irregular, Neighborhood = Moore, $p^{\text{seed}} = 0.1$ (resource density), $p^{\text{cont}} = 0.7$ (clumping density) and $n^{\text{cont}}$ (clumper pass count)

*Kernel parameters.* Within patch: $r_{\text{wp}}^{\text{min}} = 0$, $r_{\text{wp}}^{\text{max}} = 1$, $\psi_{\text{wp}} = 90$, $\hat{t}_{\text{wp}}^{\text{s}} = 30, \hat{c}_{\text{wp}}^{\text{nbd}} = 1.5$; Between patch $r_{\text{bp}}^{\text{min}} = 9$, $r_{\text{bp}}^{\text{max}} = 10$, $\psi_{\text{bp}} = 30$, $\hat{t}_{\text{bp}}^{\text{s}} = 10,000, \hat{c}_{\text{bp}}^{\text{nbd}} = 4$

*Update parameters.* $\kappa^{\text{add}} = 0.4$, $\kappa^{\text{sub}} = 0$, $\kappa^{\text{grw}} = 0.03$ and $\kappa^{\text{scl}} = 1.0$

Under pure between-patch (bp) movement on a homogeneous landscape, given that $r_{\text{bp}}^{\text{min}} = 9$ and $r_{\text{bp}}^{\text{max}} = 10$, we should expect $\overline{V}_{\text{bp}}$ after normalizing by $V_{\text{bp}} \approx 10$ to be in the range $[0.9, 1]$. Also given that the turning should, on

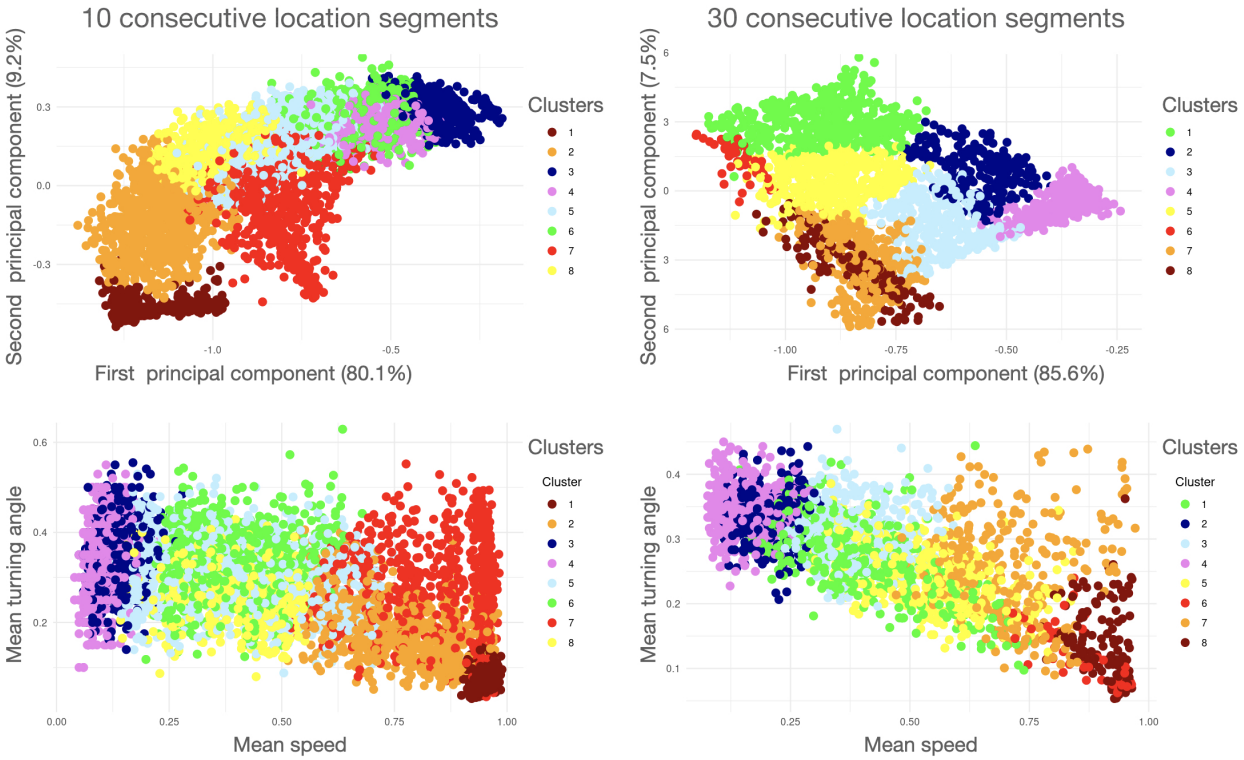


Figure 6: Results of a hierarchical cluster analysis (with number of clusters $k = 8$) performed on both 10-point (upper and lower left panels, $\nu = 10$; see Fig 3) and 30-point (upper and lower right panels, $\nu = 30$) segmentation of the simulation data generated with parameters specified for our two movement mode simulation (also see Fig C.1). The two top panels are plots of the clusters in PC1/PC2 space and the two lower panels are plots in Mean-speed/Mean-turning-angle space. A color spectrum is used to depict the smallest (blue end) to largest (red end) of segments within clusters. The centroid arguments of each cluster (Eq 15) are listed in Table 1. The colors have been selected to reflect a spectral scale of the largest to smallest mean speed for each cluster's centroid, though the cluster numbers (Column C in Table 1) are set by the clustering algorithm. Depictions of 130 randomly selected segments from each cluster for the 10-point segmentation case can be seen in Figs A.2-A.9.

Table 1: The centroid arguments (Eq 15) obtained from a hierarchical cluster analysis (with $k = 8$ clusters labeled $C = 1, \cdots, 8$, see legends in Fig 6) analysis of 10- and 30-point segmentation of our two movement mode simulation output. The results are listed in descending size of average speed ($\bar{V}$) for the centroid of each cluster. Depictions of 130 randomly selected segments from each cluster for the 10-point segmentation case can be seen in Figs A.2-A.9.

| **10-point segmentation*** | | | | | **30-point segmentation*** | | | | |
|---|---|---|---|---|---|---|---|---|---|
| C | # Segs. (%) | $\overline{V} \pm \overline{\mathrm{SD}^V}$ | $\overline{\lvert\Delta\Theta\rvert} \pm \overline{\mathrm{SD}^{\lvert\Delta\Theta\rvert}}$ | $\overline{\Delta\rho}$ | C | # Segs. (%) | $\overline{V} \pm \overline{\mathrm{SD}^V}$ | $\overline{\lvert\Delta\Theta\rvert} \pm \overline{\mathrm{SD}^{\lvert\Delta\Theta\rvert}}$ | $\overline{\Delta\rho}$ |
| 1 | 811 (11.7) | 0.95 ±0.04 | 0.08 ±0.05 | 0.86 | 8 | 138 (6.0) | 0.91 ±0.15 | 0.14 ±0.14 | 0.27 |
| 7 | 694 (10.0) | 0.84 ±0.21 | 0.28 ±0.25 | 0.14 | 6 | 51 (2.2) | 0.86 ±0.20 | 0.12 ±0.12 | 0.82 |
| 2 | 776 (11.2) | 0.75 ±0.33 | 0.18 ±0.18 | 0.79 | 7 | 306 (13.2) | 0.71 ±0.36 | 0.25 ±0.24 | 0.17 |
| 8 | 546 (7.9) | 0.38 ±0.40 | 0.27 ±0.23 | 0.81 | 5 | 416 (18.0) | 0.53 ±0.41 | 0.25 ±0.23 | 0.49 |
| 6 | 838 (12.1) | 0.38 ±0.38 | 0.33 ±0.25 | 0.11 | 1 | 463 (20.0) | 0.40 ±0.39 | 0.27 ±0.23 | 0.70 |
| 5 | 1057 (15.2) | 0.37 ±0.36 | 0.31 ±0.23 | 0.46 | 3 | 361 (15.6) | 0.39 ±0.39 | 0.31 ±0.25 | 0.14 |
| 3 | 987 (14.2) | 0.12 ±0.10 | 0.35 ±0.23 | 0.18 | 2 | 246 (10.6) | 0.20 ±0.26 | 0.33 ±0.23 | 0.29 |
| 4 | 1243 (17.9) | 0.10 ±0.07 | 0.34 ±0.23 | 0.44 | 4 | 336 (14.5) | 0.15 ±0.18 | 0.35 ±0.23 | 0.08 |

*All numbers are dimensionless due to normalizations (see text for details)

average (since $\psi_{\text{wp}} = 30$ degrees) be around $(0.5 \times 30)/180 \approx 0.083$. We carried out such a simulation over 100,000 steps and obtained $\overline{V} = 0.95$, $\overline{\text{SD}^V} = 0.03$, $\overline{|\Delta\Theta|} = 0.08$, $\overline{\text{SD}^{|\Delta\Theta|}} = 0.05$, and $\overline{\Delta^\rho} = 0.83$. These values are almost identical to the values associated with the 10-point segmentation of the data that has the largest $\overline{V}$ value or fastest speed (i.e., cluster C = 1), as reported in Table 1. Thus the segments in this cluster from our two movement mode simulation are very close to what we obtain when we simulate pure one-mode between-patch movement. The cluster with the largest $\bar{V}$ in our 30-point segmentation (cluster C = 8) does not fit nearly as well, presumably because many fewer pure bp movement segments arise in this case compared with the 10-point segmentation case.

A random selection of 130 segments from each of the 8 clusters in the case of the 10-point segmentation are plotted in Figs A.2-A.9 in Appendix A, where we note that the actual size of segments across panels is not comparable because the axes in each panel have been automatically set by our plotting routines. From these illustrations it is clear that the fastest cluster (C = 1; Fig A.2) consist primarily of unidirectional lines, as is characteristic of between-patch movement, with some small deviations and occasional changes in direction. This is in contrast with the smallest cluster (C = 4; Fig A.5) that has a number of segments switching directions by $\pi/2$ every couple of steps, as is characteristic of in-patch foraging. The difference between segments C = 6 (Fig A.7) and C = 8 (Fig A.9), which have virtually the same average speeds (Table 1) and similar average turning angles look quite different: the start and end points are relatively close with spiky profiles (C = 6) or far apart with much more open profiles (C = 8).

## 5.3 Empirical data example

We analyzed relocation data obtained from two barn owls (*Tyto alba*), using an ATLAS reverse GPS technology system that was set up in the Harod valley in northeast Israel [33]. The relocation data for both individuals, an adult female and a juvenile male (GG41259 and GG41269 in the original data set and individuals with IDs 29 and 31 in [26]), were collected a frequency of 0.25 Hz (i.e., one point every 4 seconds) during a several week period in the late summer of 2021. We used a 15-point segmentation, which corresponds to segmenting the movement track into one minute sequences. We then performed a hierarchical cluster analysis with 8 clusters, described in Appendix A.4, and obtained the results illustrated in Fig 7. The centroid statistics obtained for each cluster are provided in Table 2. Unlike the simulated results were we normalized the velocities to lie between 0 and 1, the velocities in Table 2 have the units of meters per second. Also, recall that the absolute values of average turning angles ($\overline{\Delta\Theta}$) are proportions of $\pi$ while the net displacements ($\overline{\Delta^\rho}$) are the distances between the start and end points of each segment as a proportion of the sum of the lengths of the 15 consecutive steps that make up each segment.

Table 2: The centroid arguments (Eq 15) obtained from a hierarchical cluster analysis (with $k = 8$ clusters labeled $\text{C} = 1, \cdots, 8$, see legends in Fig 6) analysis of a segmentation of tracks from two different barn owls. The results are listed in descending size of average speed ($\bar{V}$) (m/s) for the centroid of each cluster. The turning angles are proportions of $\pi$, and the displacement are proportions as described in the methods. Depictions of 130 randomly selected segments from each cluster for the Adult female and Juvenile male segmentation cases can be seen in Figs A.10-A.17 and Figs A.18-A.25 respectively.

| **Adult Female** | | | | | **Juvenile Male** | | | | |
|---|---|---|---|---|---|---|---|---|---|
| C | Segments # (%) | $\overline{V} \pm \overline{\text{SD}^V}$ (m/s) | $\overline{\lvert\Delta\Theta\rvert} \pm \overline{\text{SD}^{\lvert\Delta\Theta\rvert}}$ (units $\pi$) | $\overline{\Delta^\rho}$ (rel. #) | C | Segments # (%) | $\overline{V} \pm \overline{\text{SD}^V}$ (m/s) | $\overline{\lvert\Delta\Theta\rvert} \pm \overline{\text{SD}^{\lvert\Delta\Theta\rvert}}$ (units $\pi$) | $\overline{\Delta^\rho}$ (rel. #) |
| 8 | 4530 (5.2) | 2.80 ±1.81 | 0.24 ±0.23 | 0.79 | 8 | 3622 (5.8) | 2.55 ±1.75 | 0.24 ±0.23 | 0.77 |
| 5 | 11655 (13.3) | 1.07 ± 0.68 | 0.18 ±0.15 | 0.13 | 7 | 2679 (4.3) | 0.76 ±0.64 | 0.35 ±0.28 | 0.49 |
| 4 | 12670 (14.4) | 0.79 ± 0.52 | 0.25 ± 0.22 | 0.27 | 3 | 3553 (5.7 | 0.72 ±0.45 | 0.16 ± 0.15 | 0.09 |
| 1 | 19641 (22.4) | 0.76 ±0.47 | 0.27 ±0.25 | 0.10 | 2 | 7730 (12.4) | 0.59 ±0.40 | 0.24 ±0.21 | 0.28 |
| 6 | 9278 (10.6) | 0.71 ±0.54 | 0.33 ±0.28 | 0.42 | 4 | 9446 (15.1) | 0.51 ±0.31 | 0.44 ±0.32 | 0.08 |
| 2 | 11639 (13.3) | 0.69 ±0.42 | 0.36 ±0.31 | 0.73 | 1 | 21029 (33.6) | 0.50 ±0.30 | 0.28 ±0.24 | 0.12 |
| 3 | 10450 (11.9) | 0.61 ±0.38 | 0.39 ±0.30 | 0.22 | 5 | 5956 (9.5) | 0.50 ±0.33 | 0.41 ±0.30 | 0.29 |
| 7 | 7895 (9.0) | 0.61 ±0.36 | 0.46 ±0.31 | 0.12 | 6 | 8484 (13.6) | 0.46 ±0.27 | 0.38 ±0.29 | 0.16 |

We see from Table 2 that the average speeds of the cluster of fastest (i.e., largest) segments for the adult female and juvenile male (C=8 in both cases) are 2.80 and 2.55 m/s respectively. For both individuals, we see in Table 2 that these "fastest clusters" (as represented by $\bar{V}$) are around 3 times faster than the next fastest clusters (compare

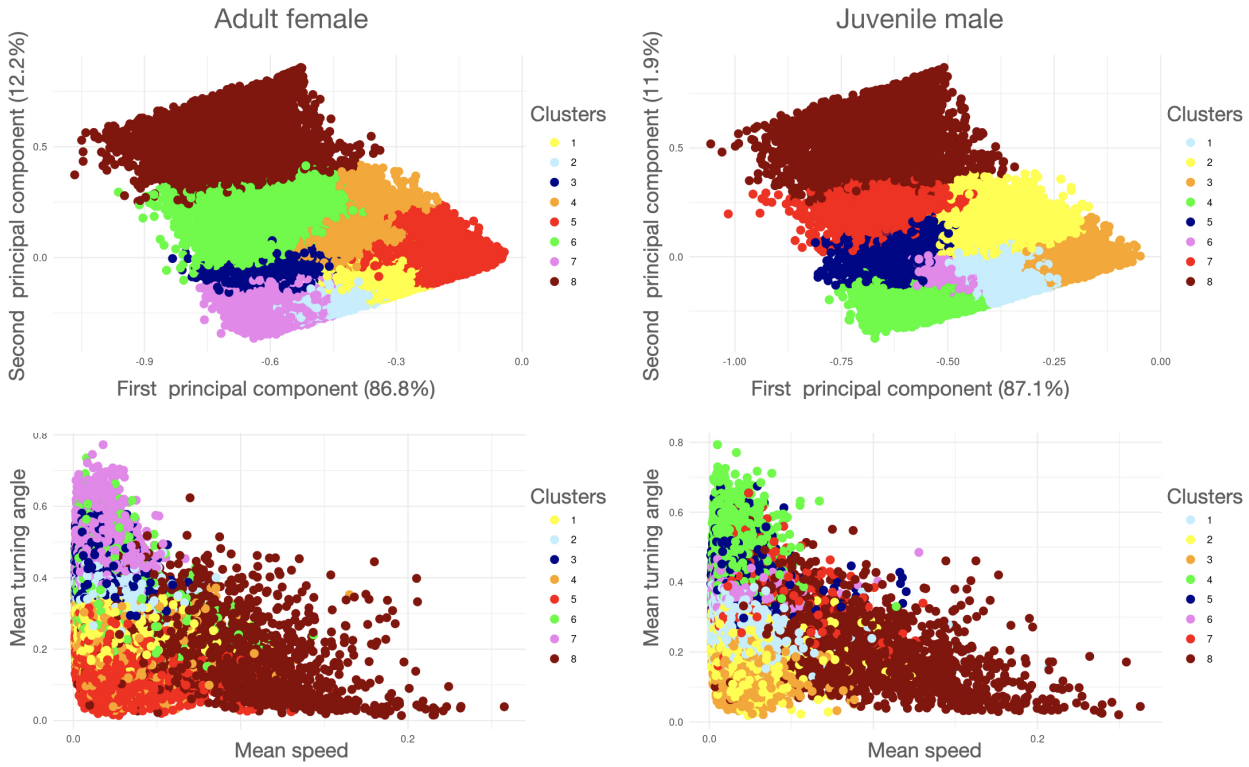


Figure 7: Results of cluster analysis ($k = 8$) performed on segmentation of the tracks of two different barn owls. As in Fig 7 The two top panels are plots of the clusters in PC1/PC2 space and the two lower panels are plots in Mean-speed/Mean-turning-angle space. A color spectrum is used to depict the smallest (blue end) to largest (red end) of segments within clusters. The centroid arguments of each cluster (Eq 15) are listed in Table 2. The colors have been selected to reflect a spectral scale of the largest to smallest mean speed for each cluster's centroid, though the cluster numbers (Column C in Table 2) are set by the clustering algorithm. Note that mean speed, as plotted here is 10 times smaller than the mean speeds $\bar{V}$ recorded in Table 2: this resizing was made to avoid extra zeros after the decimal points the in the Table. Also note that the triangular distribution of points in bottom two graphs arises because smaller turning angles are associated with the occurrence of segments with faster average speeds. Depictions of 130 randomly selected segments from each cluster for the Adult female and Juvenile male segmentation cases can be seen in Figs A.10-A.17 and Figs A.18-A.25 respectively.

first and second rows of values in Table 2), with the fastest clusters containing only 5-6% of all segments. For both individuals the bulk of the segments have speeds that lie around the geometric means of the fastest and slowest clusters of segments (i.e., between 0.35-0.8 m/s), are very similar in size but these intermediate speed clusters highly variable in their shape: some are relatively open (say, net displacement $\bar{\Delta}^{\rho} \geq 0.3$) and some are relatively closed (say, net displacement $\bar{\Delta}^{\rho} < 0.2$). The slowest clusters of the adult female and juvenile male have the relatively low speeds of 0.10 and 0.15 m/s respectively, but these slowest adult female segments are much more open on average (C = 7, $\bar{\Delta\rho} = 0.44$) than the juvenile male segments (C = 6, $\bar{\Delta\rho} = 0.08$). The reason for this requires a much closer look at the locations of these segments on the actual landscape where they occurred. Depictions of 130 randomly selected segments from each cluster for the Adult female and Juvenile male segmentation cases can be seen in Figs A.10-A.17 and Figs A.18-A.25 respectively. These depictions cannot be compared across segments for size because the axes in each panel of each figure have been automatically generated by our plotting routines. They also do not provide any landscape details. Such considerations remain a subject for future studies once analyses of the movement tracks of multiple individuals of each type have been undertaken.

As somewhat expected, a distinct correlation is evident between the mean speed and mean turning angle of a segment. This expresses itself through the lower triangular distribution of points in the mean-speed/mean-turning-

angle plots in Fig 7. This correlation also appears in the simulated data as well, but it is much less obvious: it is indicated by the negative slope of a band of points (Fig 6) rather than by a triangular distribution of points. The band implies that segments with lower speed in the simulation data have larger turning angles, while in the empirical data, lower speed still allows for a range of turning angles.

# 6 Discussion

The information that can be extracted from the movement path of an individual is akin to decoding a string of highly degraded symbols without an accompanying Rosetta stone to interpret the meaning of these symbols. In fact, interesting albeit superficial commonalities and differences can be drawn between reading a book and "reading" the life-time track of an animal's story from its birth to its death. In a book, pages are spatially well-defined objects, as are the temporally well-defined diel activity routines of the lifetime track of an animal (Fig 8). Canonical activity mode (CAM) segments within a diel activity routine (DAR) are like the sentences on a page and "activity types" represented by strings of CAM segments are like paragraphs. Unlike a book, however, where every letter is clearly visible, the letters (FuMEs) of an animal track are not at all visible, only hints of letters at regularly "spaced" (in time) intervals. Thus, instead of being able to see the words as we can in any book, we can only conjure up poor images of ersatz words in the form of StaMEs (Statistical Movement Elements, Fig 8). The book metaphor for lifetime tracks, however, is too simple in one very important way: books start with blank pages upon which letters are then printed. Tracks are generally imposed upon highly structured rather than blank landscape and the underlying landscape structure plays a decisive role in determining the spatiotemporal characteristics of the tracks that are imprinted on it. [46, 47, 48].

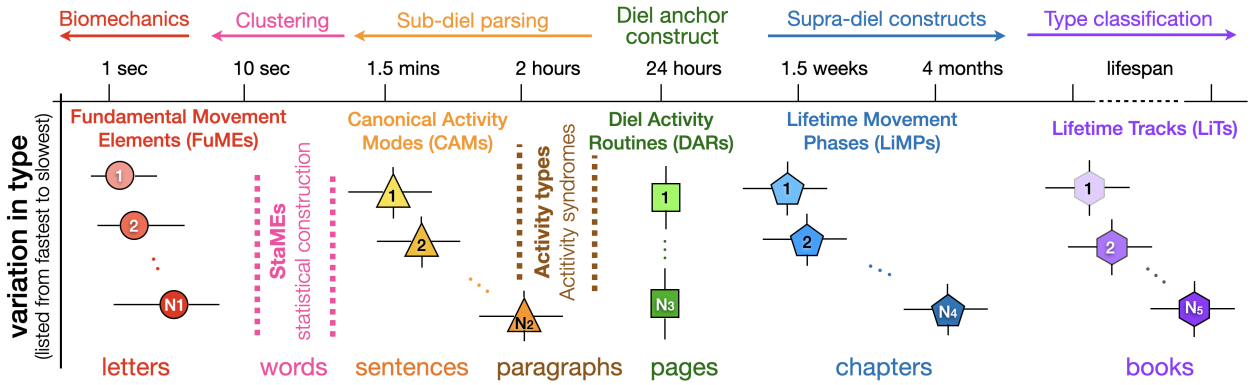


Figure 8: A facile comparison of the hierarchical movement track (or walk $W$—see Eq 12) segmentation used to help tell the story of an animal's life history through its movement path and reading a story in a book (adapted from Fig 1 in [15]). Some $N_1$ types of **FuMEs** (letters) are not identifiable from track data. **StaMEs** (words) can be constructed and classified using the methods discussed in this paper. Some $N_2$ types of **CAMs** (sentences) are sequences of StaMEs of the same type, while **Activity types** (paragraphs) are characteristic mixtures of CAMs. Some $N_3$ types of **DARs** (pages) can be identified using spatial or geometric measures of 24 hour tracks [26], as can some $N_4$ types of LiMPs (chapters) and $N_5$ types of **LiTTs** (books) [49] (also see Appendix A.2).

In some ways, our Rosetta stone is our simulation model. It allows us through simulations to see how different kernels produce segments with particular sets of statistics. We illustrated this in Section 5.1 where we showed, for example, that a kernel with minimum- and maximum-step-length and range-of-turning-angle parameters values $r^{\min} = 0$, $r^{\max} = 5$, and $\psi = \pi/2$ creates a cluster of segments whose centroid has mean step-length and turning-angles (standard deviations in parenthesis) of 3.38 ($\pm$1.23) and $0.27\pi$ ($\pm 0.15\pi$) (Eq 17).

More ambitiously, we could numerically construct 2-patch mappings, as defined in Eq 16, where Table 2 represents the set of centroids $\mathscr{X}^{\text{patch}}_{0,\mu,k}$ (Eq 15) obtained under a hierarchical 8-cluster, $\mu = 15$ point-segmentation of the simulated data (using $\texttt{RAM}^{\text{patch}}_0$) when the arguments of this mapping are (see Eq 17) the simulation parameters listed in Section 5.2. In the case of trying to match movement modes to empirical relocation data, we can use the actual landscape

associated with those data, plus set selection functions fitted to different parts of the landscape (e.g., movement within and between areas where particular types of activities take place). This appears to be a challenging process where it remains to be seen in future studies how the methods discussed in this paper can be applied to identifying putative movement modes that appear to be used by individuals when moving across real landscapes and carrying out a range of activities.

Methods will have to be developed to elucidate the role that the within-movement-mode time parameters such as $\hat{t}^s_\alpha$ (in 2-patch case $\alpha =$ wp or bp) play in determining the lengths of pure wp or bp (e.g., see Fig C.1). Clearly, this would require extensive computations to make progress with such a research program. Additionally, relocation data may be reinforced by other kinds of data obtained, for example, through remote sensing (e.g., satellites or drones) or behavioral classifications obtained from accelerometer data [50, 51] to help extract such hidden or subtle aspects of movement behavior.

Of course, other approaches to simulating movement have been developed, several of which are mathematically more sophisticated than the approach we take. Some of these treat movement as continuous-time stochastic process for which Brownian motion (purely random movement) and Orstein-Uhlenbeck processes (directionally correlated random walks) and modifications thereto (e.g., velocity correlated walks, central attractors, etc.) [52, 53, 54, 55] are examples. Others represent movement in terms of partial differential equations some random components that switch between gradient following search (advection plus random noise) and random search [56] or discrete approaches using integrated step-selection functions [57].

It is worth noting that spatio-temporally continuous models anyway require discretization schemes to generate solutions, but the aim is to keep the discretization small enough and solution methods robust enough so the associated discretization errors do not affect interpretations of outcomes. Formulating movement models discretely, as we have, with regards to space and time avoids the issue of what is an appropriate solution method, but introduces both philosophical and practical issues regarding differences in discrete- versus continuous-time representations of ecological processes [58], particularly since data always discretized in one way or another [59]. Additionally, discrete time, space, and variable trait formulations of ecological processes facilitate the incorporation of idiosyncratic details that elude the more rigorous formulation of continuous variable formulations. This is particularly true when it comes to turning model formulations into computational code for analysis through numerical simulation. Such formulations are naturally facilitated when movement is embedded into network structures [60, 61] or is modeled as taking place on cellular arrays [62, 63, 64, 65].

Given the complex shapes of the segments plotted in Figs A.2-A.25, the hierarchical clustering approach that we have taken to parsing the movement track segments into categories may not be the best approach. Supervised and unsupervised deep-learning approaches provide less forced and more powerful ways to extract clusters, particularly those targeted at time series data [66, 67, 68, 69, 70, 71]. In terms of supervised, deep-learning approaches, convolutional neural nets (CNNs) of appropriate types may be trained using segments simulated by ANIMOVER_1 over real landscapes, obtained through the application of various sets of movement rules, as a way of generating the training sets [72]. Once trained these CNNs can then be used to classify empirical data segments thereby providing some insights into the types of movement rules that may have been used by actual individuals when moving over the landscapes in question.

In the mean time, insights into the ecological underpinnings of the movement tracks can be obtained by studying them at other hierarchical levels. As indicated in Fig 8, the diel activity routine (DAR) is perhaps one of the easier levels to study [26]. The reason for this is its fixed temporal length of 24 hours, which makes it possible to segment in a relatively unambiguous way. The only ambiguity is what point in time during the diel period should be fixed as the start/end point for each DAR. This problem has been discussed elsewhere [73, 74]. Additionally, DARs can be strung together to obtain supra-diel constructs (Fig 8) such as the extent of seasonal range of individuals [75, 17], and beyond this to classifying the syndromic movement behavior from whole or abbreviated lifetime tracks [49].

# 7 Conclusion

Over the next decade we can expect exponential increases over time in the availability of high-frequency movement relocation time series for many different species, reflecting both growing interest in the topic, and technological improvements in tracking methods. This increase, together with the increasing computational power of server clusters, the use

of parallel processing, and improvements and extensions to our ANIMOVER_1 simulator will make the methods discussed in this paper for deconstructing diel activity routine tracks into StaMEs more reliable and easier to implement. Additionally, deep learning methods and possibly some other machine learning methods [72, 68, 69, 70, 71, 76] that are able to account for complexities in the shapes of the segments arising from our analysis (e.g., see Figs A.2-A.25) may be more useful for categorizing segments than the hierarchical clustering approach used here. It will also facilitate using StaMEs to bring movement canonical activity modes and higher level behavioral activities or syndromes [77, 17] into sharper focus using various time series forecasting techniques [78, 79] and more sophisticated movement simulation algorithms.

## Acknowledgments

Assaf Uzan, Michal Handel, Nadav Amir, Shani Cain and Yohay Wasserlauf helped with fieldwork and other components of data collection. We are also grateful to many people and institutes who permit us to use their towers and power supply for our base stations. These include Eli Chupa and Pablo Uner (Harod Valley Water Supply Co-op LTD), Shmuel Wirzberger (Kibbutz Beit Hashita), Eytan Ofer (Kibbutz Sde Eliyahu), Yitzhak Rivlin, Yado Shalev and Elon Shaked (Kibbutz Ein HaNetziv); Shaldag Refrigeration Storeroom LTD, Avihu Levanon (Partner PHI LTD), Michael Zelmanov (Pelephone Communications Ltd), Asher Nahmani and Rafi Levi (Mekorot), and Eli Cohen and Eli Mimran (Bezeq Inc.), and The Israeli Telecommunication Corp Ltd.

## Funding

This work was funded in part by the A. Starker Leopold Chair of Wildlife Ecology at UC Berkeley, by the Minerva Foundation, the Minerva Center for Movement Ecology, and grants ISF-965/15, 1919/19 and 396/20, from the Israel Science Foundation. Further, support was given by a grant from the Data Science Center at Tel Aviv University (TAD), and by the Koret-UC Berkeley-Tel Aviv University Initiative in Computational Biology and Bioinformatics (WMG, OS, ST). SC was also supported by the Hoopoe Foundation, Society for the Protection of Nature in Israel, Ministry of Agriculture and Rural Development, Ministry of Regional Cooperation, Larry Kornhauser, and Peter and Naomi Neustadter.

## Authors contribution

WMG designed the study, formulated the model, and drafted the manuscript
RS built the Numerus coding platform and RAMP simulators
VS carried out the StaME extraction and clustering procedures
OS contributed to model design
SC, RN, OS, and ST were involved in the setting up and running the ATLAS system and collecting and organizing the barn owl relocation data
All authors read and edited various versions of the manuscript including the final version

## Glossary

For the convenience of the reader we provide a glossary of indices and symbols in Table 3.

Table 3: **Indices and symbols used to formulate the model's structure**

| Symbols | Explanation | Ref. |
|---|---|---|
| *Indices and periods* | | |
| $a = 1, \cdots, n^{\text{row}}$<br>$b = 1, \cdots, n^{\text{col}}$ | Row and column indices to denote landscape cell$(a,b)$ | Eq 1 |
| $t = 0, \cdots, n^{\text{time}}$, $t^{\text{stop}}$ | Index and number of time steps in global simulation, stop time if agent value drops to 0 | Fig 2 |
| $\mu$, $z = 1, \cdots, n_{\text{seg}}$ | Number of consecutive points in a StaME segment, index for and number of segments in a DAR | Fig 3 |
| $\alpha = 1, \cdots, n^{\text{fume}}$ | StaME index and number | Eq 4 |
| $\lambda = 1, \cdots, n^{\text{rule}}$, $\mathscr{R}_{\lambda}$, $\mathscr{R}_{\lambda_\alpha}$ | Rule index and number, step-selection rules, $\alpha$-specific rule | Eq 5 |
| $\iota, \cdots, k$ | Cluster index (iota) and number of clusters | Eq 15 |
| *Structures* | | |
| $(x_a, y_b)$, $\Delta x$, $\Delta y$, $\Omega_{\text{scale}}$ | Locations of cell$(a,b)$ on the landscape, row and column distances between consecutive cells, and the time-space scaling constant | |
| $\mathscr{A}(t)$, $c_{ab,t}$, $(x_t, y_t)$, $\theta_t$ | The state of cellular array $\mathscr{A}$, of cell$_{ab}$, and location and angle of heading of the agent at time $t$ | Eq 1 |
| $p^{\text{seed}}$, $p^{\text{cont}}$, $n^{\text{cont}}$ | Parameters used to generate a patchy landscape | $\texttt{RAM}^{\text{patch}}$ |
| $\rho_{ab}(x_t, y_t)$, $\theta_{ab}(x_t, y_t)$ | Distance between the individuals current location and a cell $(x_a, y_b)$ and angle of heading to this cell | Eq 2-3 |
| $K_\alpha$, $\mathscr{K}$, $r_\alpha^{\text{min}}$, $r_\alpha^{\text{max}}$, $\psi_\alpha$ | StaME kernels, kernel $\alpha$ and its min & max SL (step length, aka velocity), and TA (turning-angle) range arguments | Eq 4 |
| $\tilde{K}_t^\alpha(\text{args})$, $\theta_t$ $\mathscr{R}_{\lambda_\alpha}$,<br>$\hat{t}_s$, $\hat{t}_\alpha^{\text{s}}$, $\hat{c}_\alpha^{\text{nbh}}$ | Movement kernel $\alpha$ at time $t$ and angle of heading angle of heading, step-selection rule, time in current mode, mode switching and neighborhood-value parameters | Eq 5 |
| $h_t$, $t_t^{\text{s}}$, $h_{\text{next}}$, $c_{cb,\text{next}}$ | Agent value at time $t$, time spent in current movement mode, agent and cell interim values | Eq 6-8 |
| $\kappa^{\text{add}}$, $\kappa^{\text{sub}}$, $\kappa^{\text{grw}}$, $\kappa^{\text{scl}}$ | Parameter values used in cell and agent updating equations | Eq 6-7 |
| $\mathscr{C}_t^\alpha(x_t, y_t, \theta_t)$ | The set of all cells that overlap with kernel $\tilde{K}_t^\alpha(\text{args})$ | Eq 9 |
| $\mathscr{P}_t^\alpha(x_t, y_t, \theta_t)$ | The set of probability values for cells that overlap with kernel $\tilde{K}_t^\alpha(\text{args})$ | Eq 10 |
| $p_\alpha(t_s)$, $\hat{t}_\alpha^s$ | Probability that an individual in mode $\alpha$ for period $t^s$ continues in this mode, the $\alpha$ mode switching function parameter value | Eq 11 |
| $W$, $V$, $\Delta\Theta$ | Stochastic walk, velocity (aka step length:SL) and turning-angle (TA) relocation time series | Eqs 12, A.1-A.3 |
| $\text{Seg}_z$, $\mathscr{S}_\mu$, $V_z$, $\text{SD}_z^V$, $\lvert\Delta\Theta\rvert_z$,<br>$\text{SD}_z^{\lvert\Delta\Theta\rvert}$, $\Delta_z^\rho$ | Segment $z$, set of segments, segment specific average velocity and standard deviation, average turning angle and standard deviation, net displacement | Eq 13 |
| $\mathscr{F}_\mu^{\text{hom}}$ | Mapping of the parameters of kernel $K_\alpha$ onto the centroid of all segments generated by a walk using this kernel on a homogeneous landscape | Eq 14 |
| $\mathscr{X}_{0,\mu,k}^{patch}$, $\mathscr{F}_{0,\mu,k}^{patch}(\text{args})$ | Set of centroids from a hierarchical cluster analysis of a 2-mode patchy landscape simulation using $\texttt{RAM}_0^{\text{patch}}$, and resulting discrete mapping from the relevant parameter space to the centroid space | Eq 15-16 |

# References


[1] Eric R Dougherty, Dana P Seidel, Colin J Carlson, Orr Spiegel, and Wayne M Getz. Going through the motions: incorporating movement analyses into disease research. *Ecology letters*, 21(4):588–604, 2018.

[2] David Saltz and Wayne M Getz. Finding a home: Stopping theory and its application to home range establishment in a novel environment. *Frontiers in Conservation Science*, 2:714580, 2021.

[3] Briana Abrahms, Ellen O Aikens, Jonathan B Armstrong, William W Deacy, Matthew J Kauffman, and Jerod A Merkle. Emerging perspectives on resource tracking and animal movement ecology. *Trends in Ecology & Evolution*, 36(4):308–320, 2021.

[4] Andrew M Allen and Navinder J Singh. Linking movement ecology with wildlife management and conservation. *Frontiers in Ecology and Evolution*, 3:155, 2016.

[5] Oded Berger-Tal and David Saltz. Invisible barriers: anthropogenic impacts on inter-and intra-specific interactions as drivers of landscape-independent fragmentation. *Philosophical Transactions of the Royal Society B*, 374 (1781):20180049, 2019.

[6] Sandra A Binning, Allison K Shaw, and Dominique G Roche. Parasites and host performance: incorporating infection into our understanding of animal movement. *Integrative and Comparative Biology*, 57(2):267–280, 2017.

[7] Allison K Shaw. Causes and consequences of individual variation in animal movement. *Movement ecology*, 8 (1):12, 2020.

[8] Jonathan B Armstrong, Gaku Takimoto, Daniel E Schindler, Matthew M Hayes, and Matthew J Kauffman. Resource waves: phenological diversity enhances foraging opportunities for mobile consumers. *Ecology*, 97(5): 1099–1112, 2016.

[9] Kasper Thorup, Anders P Tøttrup, Mikkel Willemoes, Raymond HG Klaassen, Roine Strandberg, Marta Lomas Vega, Hari P Dasari, Miguel B Araújo, Martin Wikelski, and Carsten Rahbek. Resource tracking within and across continents in long-distance bird migrants. *Science Advances*, 3(1):e1601360, 2017.

[10] Briana Abrahms, Elliott L Hazen, Ellen O Aikens, Matthew S Savoca, Jeremy A Goldbogen, Steven J Bograd, Michael G Jacox, Ladd M Irvine, Daniel M Palacios, and Bruce R Mate. Memory and resource tracking drive blue whale migrations. *Proceedings of the National Academy of Sciences*, 116(12):5582–5587, 2019.

[11] Juan M Morales, Paul R Moorcroft, Jason Matthiopoulos, Jacqueline L Frair, John G Kie, Roger A Powell, Evelyn H Merrill, and Daniel T Haydon. Building the bridge between animal movement and population dynamics. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365(1550):2289–2301, 2010.

[12] Graeme C Hays, Helen Bailey, Steven J Bograd, W Don Bowen, Claudio Campagna, Ruth H Carmichael, Paolo Casale, Andre Chiaradia, Daniel P Costa, Eduardo Cuevas, et al. Translating marine animal tracking data into conservation policy and management. *Trends in ecology & evolution*, 34(5):459–473, 2019.

[13] Susanne H Sokolow, Nicole Nova, Kim M Pepin, Alison J Peel, Juliet RC Pulliam, Kezia Manlove, Paul C Cross, Daniel J Becker, Raina K Plowright, Hamish McCallum, et al. Ecological interventions to prevent and manage zoonotic pathogen spillover. *Philosophical Transactions of the Royal Society B*, 374(1782):20180342, 2019.

[14] Simon Benhamou. Of scales and stationarity in animal movements. *Ecology letters*, 17(3):261–272, 2014.

[15] Wayne M Getz. A hierarchical path-segmentation movement ecology framework. *Ecological Processes*, 11(1): 1–15, 2022.

[16] Wayne Marcus Getz. An animal movement track segmentation framework for forecasting range adaptation under global change. *Frontiers in Ecology and Evolution*, 11:1171169, 2023.

[17] Roland Kays, Ben Hirsch, Damien Caillaud, Rafael Mares, Shauhin Alavi, Rasmus Worsøe Havmøller, and Margaret Crofoot. Multi-scale movement syndromes for comparative analyses of animal movement patterns. *Movement Ecology*, 11(1):61, 2023.

[18] Nathalie Gontier. Hierarchies, networks, and causality: the applied evolutionary epistemological approach. *Journal for General Philosophy of Science*, 52(2):313–334, 2021.

[19] Baptiste Garde, Rory P Wilson, Adam Fell, Nik Cole, Vikash Tatayah, Mark D Holton, Kayleigh AR Rose, Richard S Metcalfe, Hermina Robotka, Martin Wikelski, et al. Ecological inference using data from accelerometers needs careful protocols. *Methods in Ecology and Evolution*, 13(4):813–825, 2022.

[20] Tessa A Van Walsum, Andrea Perna, Charles M Bishop, Campbell P Murn, Philip M Collins, Rory P Wilson, and Lewis G Halsey. Exploring the relationship between flapping behaviour and accelerometer signal during ascending flight, and a new approach to calibration. *Ibis*, 162(1):13–26, 2020.

[21] Roland Kays, Margaret C Crofoot, Walter Jetz, and Martin Wikelski. Terrestrial animal tracking as an eye on life and planet. *Science*, 348(6240):aaa2478, 2015.

[22] CE Beardsworth, E Gobbens, F van Maarseveen, B Denissen, A Dekinga, R Nathan, S Toledo, and AI Bijleveld. Validating atlas: a regional-scale, high-throughput tracking system. *Methods Molecular Evolution*, 2022.

[23] Laila D Kazimierski, Nicolás E Catalano, Karina Laneri, Agustina Balazote Oliver, Gabriela Calzolari, Julien Joseph, Guillermo C Amico, and Guillermo Abramson. Trajectory assessment of the vulnerable marsupial dromiciops gliroides in the patagonian temperate forest. *Mammalian Biology*, pages 1–13, 2021.

[24] Eliezer Gurarie, Russel D Andrews, and Kristin L Laidre. A novel method for identifying behavioural changes in animal movement data. *Ecology letters*, 12(5):395–408, 2009.

[25] N Owen-Smith and V Goodall. Coping with savanna seasonality: comparative daily activity patterns of a frican ungulates as revealed by gps telemetry. *Journal of Zoology*, 293(3):181–191, 2014.

[26] Ludovica Luisa Vissat, Shlomo Cain, Sivan Toledo, Orr Spiegel, and Wayne M Getz. Categorizing the geometry of animal diel movement patterns with examples from high-resolution barn owl tracking. *Movement Ecology*, 11 (1):1–20, 2023.

[27] N Sören Häfker and Kristin Tessmar-Raible. Rhythms of behavior: are the times changin'? *Current Opinion in Neurobiology*, 60:55–66, 2020.

[28] Navinder J Singh, Luca Börger, Holger Dettki, Nils Bunnefeld, and Göran Ericsson. From migration to nomadism: movement variability in a northern ungulate across its latitudinal range. *Ecological Applications*, 22 (7):2007–2020, 2012.

[29] Robert S Schick, Scott R Loarie, Fernando Colchero, Benjamin D Best, Andre Boustany, Dalia A Conde, Patrick N Halpin, Lucas N Joppa, Catherine M McClellan, and James S Clark. Understanding movement data and movement processes: current and emerging directions. *Ecology letters*, 11(12):1338–1350, 2008.

[30] EA Codling and NA Hill. Sampling rate effects on measurements of correlated and biased random walks. *Journal of Theoretical Biology*, 233(4):573–588, 2005.

[31] Juan Manuel Morales, Daniel T Haydon, Jacqui Frair, Kent E Holsinger, and John M Fryxell. Extracting more out of relocation data: building movement models as mixtures of random walks. *Ecology*, 85(9):2436–2445, 2004.

[32] Uri Wilensky and William Rand. *An introduction to agent-based modeling: modeling natural, social, and engineered complex systems with NetLogo*. Mit Press, 2015.

[33] Shlomo Cain, Tovale Solomon, Yossi Leshem, Sivan Toledo, Eitam Arnon, Alexandre Roulin, and Orr Spiegel. Movement predictability of individual barn owls facilitates estimation of home range size and survival. *Movement Ecology*, 11(1):10, 2023.

[34] Ammon Corl, Motti Charter, Gabe Rozman, Sivan Toledo, Sondra Turjeman, Pauline L Kamath, Wayne M Getz, Ran Nathan, and Rauri CK Bowie. Movement ecology and sex are linked to barn owl microbial community composition. *Molecular ecology*, 29(7):1358–1371, 2020.

[35] Wayne M Getz. Biomass transformation webs provide a unified approach to consumer–resource modelling. *Ecology letters*, 14(2):113–124, 2011.

[36] Henrik Thurfjell, Simone Ciuti, and Mark S Boyce. Applications of step-selection functions in ecology and conservation. *Movement ecology*, 2(1):4, 2014.

[37] Thierry Duchesne, Daniel Fortin, and Louis-Paul Rivest. Equivalence between step selection functions and biased correlated random walks for statistical inference on animal movement. *PloS one*, 10(4):e0122947, 2015.

[38] Manuela Panzacchi, Bram Van Moorter, Olav Strand, Marco Saerens, Ilkka Kivimäki, Colleen C St. Clair, Ivar Herfindal, and Luigi Boitani. Predicting the continuum between corridors and barriers to animal movements using step selection functions and randomized shortest paths. *Journal of Animal Ecology*, 85(1):32–42, 2016.

[39] Tal Avgar, Jonathan R Potts, Mark A Lewis, and Mark S Boyce. Integrated step selection analysis: bridging the gap between resource selection and animal movement. *Methods in Ecology and Evolution*, 7(5):619–630, 2016.

[40] Haiganoush K Preisler, Alan A Ager, and Michael J Wisdom. Analyzing animal movement patterns using potential functions. *Ecosphere*, 4(3):1–13, 2013.

[41] Pratik Rajan Gupte, Christine E Beardsworth, Orr Spiegel, Emmanuel Lourie, Sivan Toledo, Ran Nathan, and Allert I Bijleveld. A guide to pre-processing high-throughput animal tracking data. *Journal of Animal Ecology*, 91(2):287–307, 2022.

[42] Thomas Wolf and Robert Konrath. Avian wing geometry and kinematics of a free-flying barn owl in flapping flight. *Experiments in Fluids*, 56:1–18, 2015.

[43] Jialei Song. Fly low: The ground effect of a barn owl (tyto alba) in gliding flight. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 235(2):308–318, 2021.

[44] Wayne M Getz, Ludovica Luisa Vissat, and Richard Salter. Simulation and analysis of animal movement paths using numerus model builder. In *2020 Spring Simulation Conference (SpringSim)*, pages 1–12. IEEE, 2020.

[45] Wayne M Getz, Richard Salter, and Ludovica Luisa Vissat. Simulation applications to support teaching and research in epidemiological dynamics. *BMC medical education*, 22(1):632, 2022.

[46] Tim S Doherty, Charles N Fist, and Don A Driscoll. Animal movement varies with resource availability, landscape configuration and body size: a conceptual model and empirical example. *Landscape Ecology*, 34:603–614, 2019.

[47] Nicolas Lubitz, Michael Bradley, Marcus Sheaves, Neil Hammerschlag, Ryan Daly, and Adam Barnett. The role of context in elucidating drivers of animal movement. *Ecology and Evolution*, 12(7):e9128, 2022.

[48] Vanessa Brum-Bastos, Marcelina Łoś, Jed A Long, Trisalyn Nelson, and Urška Demšar. Context-aware movement analysis in ecology: a systematic review. *International Journal of Geographical Information Science*, 36 (2):405–427, 2022.

[49] Briana Abrahms, Dana P Seidel, Eric Dougherty, Elliott L Hazen, Steven J Bograd, Alan M Wilson, J Weldon McNutt, Daniel P Costa, Stephen Blake, Justin S Brashares, et al. Suite of simple metrics reveals common movement syndromes across vertebrate taxa. *Movement ecology*, 5(1):1–11, 2017.

[50] Ran Nathan, Orr Spiegel, Scott Fortmann-Roe, Roi Harel, Martin Wikelski, and Wayne M Getz. Using tri-axial acceleration data to identify behavioral modes of free-ranging animals: general concepts and tools illustrated for griffon vultures. *Journal of Experimental Biology*, 215(6):986–996, 2012.

[51] Orr Spiegel, Roi Harel, Alejandro Centeno-Cuadros, Ohad Hatzofe, Wayne M Getz, and Ran Nathan. Moving beyond curve fitting: using complementary data to assess alternative explanations for long movements of three vulture species. *The American Naturalist*, 185(2):E44–E54, 2015.

[52] Justin M Calabrese, Chris H Fleming, and Eliezer Gurarie. ctmm: an r package for analyzing animal relocation data as a continuous-time stochastic process. *Methods in Ecology and Evolution*, 7(9):1124–1132, 2016.

[53] Ionut Paun, Dirk Husmeier, J Grant C Hopcraft, Majaliwa M Masolele, and Colin J Torney. Inferring spatially varying animal movement characteristics using a hierarchical continuous-time velocity model. *Ecology Letters*, 25(12):2726–2738, 2022.

[54] Ohad Vilk, Yotam Orchan, Motti Charter, Nadav Ganot, Sivan Toledo, Ran Nathan, and Michael Assaf. Ergodicity breaking in area-restricted search of avian predators. *Physical Review X*, 12(3):031005, 2022.

[55] Ohad Vilk, Erez Aghion, Ran Nathan, Sivan Toledo, Ralf Metzler, and Michael Assaf. Classification of anomalous diffusion in animal movement data using power spectral analysis. *Journal of Physics A: Mathematical and Theoretical*, 55(33):334004, 2022.

[56] William F Fagan, Tyler Hoffman, Daisy Dahiya, Eliezer Gurarie, Robert Stephen Cantrell, and Chris Cosner. Improved foraging by switching between diffusion and advection: benefits from movement that depends on spatial context. *Theoretical ecology*, 13:127–136, 2020.

[57] Johannes Signer, John Fieberg, Björn Reineking, Ulrike Schlägel, B. Smith, Niko Balkenhol, and Tal Avgar. Simulating animal space use from fitted integrated step-selection functions ( issf ). *Methods in Ecology and Evolution*, 12 2023. doi: 10.1111/2041-210X.14263.

[58] Wayne M Getz. An introspection on the art of modeling in population ecology. *BioScience*, 48(7):540–552, 1998.

[59] Annick Lesne. The discrete versus continuous controversy in physics. *Mathematical Structures in Computer Science*, 17(2):185–223, 2007.

[60] Guillaume Bastille-Rousseau, Iain Douglas-Hamilton, Stephen Blake, Joseph M Northrup, and George Wittemyer. Applying network theory to animal movements to identify properties of landscape space use. *Ecological Applications*, 28(3):854–864, 2018.

[61] Shenglai Yin, Yanjie Xu, Mingshuai Xu, Mart CM de Jong, Mees RS Huisman, Andrea Contina, Herbert HT Prins, Zheng YX Huang, and Willem F de Boer. Habitat loss exacerbates pathogen spread: An agent-based model of avian influenza infection in migratory waterfowl. *PLoS Computational Biology*, 18(8):e1009577, 2022.

[62] Iain D Couzin, Jens Krause, Nigel R Franks, and Simon A Levin. Effective leadership and decision-making in animal groups on the move. *Nature*, 433(7025):513, 2005.

[63] Wayne M Getz, Richard Salter, Andrew J Lyons, and Nicolas Sippl-Swezey. Panmictic and clonal evolution on a single patchy resource produces polymorphic foraging guilds. *PloS one*, 10(8):e0133732, 2015.

[64] Wayne M Getz, Richard Salter, Dana Paige Seidel, and Pim Van Hooft. Sympatric speciation in structureless environments. *BMC evolutionary biology*, 16:1–12, 2016.

[65] Jan Salecker, Marco Sciaini, Katrin M Meyer, and Kerstin Wiegand. The nlrx r package: A next-generation framework for reproducible netlogo model analyses. *Methods in Ecology and Evolution*, 10(11):1854–1863, 2019.

[66] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR, 2016.

[67] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963, 2019.

[68] Md Rezaul Karim, Oya Beyan, Achille Zappa, Ivan G Costa, Dietrich Rebholz-Schuhmann, Michael Cochez, and Stefan Decker. Deep learning-based clustering approaches for bioinformatics. *Briefings in bioinformatics*, 22(1):393–415, 2021.

[69] Baptiste Lafabregue, Jonathan Weber, Pierre Gançarski, and Germain Forestier. End-to-end deep representation learning for time series clustering: a comparative study. *Data Mining and Knowledge Discovery*, 36(1):29–81, 2022.

[70] Yazhou Ren, Jingyu Pu, Zhimeng Yang, Jie Xu, Guofeng Li, Xiaorong Pu, Philip S Yu, and Lifang He. Deep clustering: A comprehensive survey. *arXiv preprint arXiv:2210.04142*, 2022.

[71] Óscar Jiménez Rama, Fernando Moreno-Pino, David Ramírez, and Pablo M Olmos. Interpretable spectral variational autoencoder (isvae) for time series clustering. *arXiv e-prints*, pages arXiv–2310, 2023.

[72] Hao Wu, Qi Liu, and Xiaodong Liu. A review on deep learning approaches to image classification and object segmentation. *Computers, Materials & Continua*, 60(2), 2019.

[73] Dana Paige Seidel, Wayne L Linklater, Werner Kilian, Pierre du Preez, and Wayne M Getz. Mesoscale movement and recursion behaviors of namibian black rhinos. *Movement ecology*, 7(1):1–14, 2019.

[74] Norman Owen-Smith. Daily movement responses by african savanna ungulates as an indicator of seasonal and annual food stress. *Wildlife Research*, 40(3):232–240, 2013.

[75] Norman Owen-Smith and Jodie Martin. Identifying space use at foraging arena scale within the home ranges of large herbivores. *PloS one*, 10(6):e0128821, 2015.

[76] Gorka Muñoz-Gil, Miguel Angel Garcia-March, Carlo Manzo, José D Martín-Guerrero, and Maciej Lewenstein. Single trajectory characterization via machine learning. *New Journal of Physics*, 22(1):013010, 2020.

[77] Orr Spiegel, Stephan T. Leu, C. Michael Bull, and Andrew Sih. What's your move? movement as a link between personality and spatial dynamics in animal populations. *Ecology Letters*, 20(1):3–18, 2017. doi: https://doi.org/10.1111/ele.12708. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/ele.12708.

[78] José F Torres, Dalil Hadjout, Abderrazak Sebaa, Francisco Martínez-Álvarez, and Alicia Troncoso. Deep learning for time series forecasting: a survey. *Big Data*, 9(1):3–21, 2021.

[79] Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.

[80] Daniel Müllner. fastcluster: Fast hierarchical, agglomerative clustering routines for r and python. *Journal of Statistical Software*, 53(9):1–18, 2013.

[81] Adi L Tarca, Vincent J Carey, Xue-wen Chen, Roberto Romero, and Sorin Drăghici. Machine learning and its applications to biology. *PLoS computational biology*, 3(6):e116, 2007.

[82] Anne Thessen. Adoption of machine learning techniques in ecology and earth science. *One Ecosystem*, 1:e8621, 2016.

[83] John Joseph Valletta, Colin Torney, Michael Kings, Alex Thornton, and Joah Madden. Applications of machine learning in animal behaviour studies. *Animal Behaviour*, 124:203–220, 2017.

# Appendices

These appendices contain references to equations and tables appearing in the main text of *Getz et al., The Statistical Building Blocks of Animal Movement Simulations* and the bibliographic citations herein are included in the *References* section of that text.

## A Hierarchical Segmentation and Empirical Data

### A.1 Issues of scale

We have various temporal scales in our model as they relate to the following structures and processes from likely fastest to slowest:

1 FuME scale (fundamental movement element). The scale of the actual "hidden" movement elements underlying the StaMEs (previously called metaFuMEs). This is the average time it takes to perform a typical FuME, such as one repeatable sequence when striding, wing flapping, wiggling or undulating one's body.

2 Landscape traversal scale. This converts the landscape pixel scale to the typical time it takes for individual to move across one pixel. This scale is typically finer than the relocation data scale (see next), though if the pixels are large (e.g., 10 meters or more) and the relocation data scale fine then this scale may be coarser than the relocation data scale.

3 Relocation data scale. This is set by the frequency of the relocation data. Fine scale implies that data points are less than 10 secs apart. Course scale implies points more than 1 minute apart. Intermediate scale is between these two.

4 STaMe scale (statistical movement elements, formerly metaFuMEs). This is the scale of our StaME elements—i.e., it relates to the number of relocation data points (including missing points) used to compute our StaMEs, which my vary from typically 10 to 30 points, depending on the fineness of the relocation data scale.

5 CAM scale (Canonical activity mode) (formerly short or homogeneous CAMs). This scale relates to the typical length of a homogeneous sequence of some underlying metaFuMEs. This scale will vary somewhat, depending on the type of behavior. Directed walking, for example, may last several hours, though it will likely be interspersed with pauses when the individual rests or scans their surrounding for information relating to navigation or safety. Grazing, which is a mix of several different FuMEs that each last a short time (e.g., stop-start motions involving forward, sideways and backward movements coupled with various head motions) may also last on the order of large fractions of an hour or multiple hours.

6 Activity type scale (formerly long CAMs). When relocation data is relatively fine (consecutive points are seconds apart) then "activity types" may be identified in terms of a characteristic mix of CAMs. Thus a migration movement CAM that lasts for a substantial part of a diel may be made up of a characteristic mix of directed movement, grazing, and resting CAMs. On the other hand, if the relocation data scale is relatively course (i.e., relocation points are minutes apart) then StaMEs reflect average movement behavior during "migration movement" may already have statistics that reflect a mix of directed movement, grazing, and resting behaviors.

### A.2 Movement path data and segmentation

Here is a refinement of the hierarchical definitions provided in [15] to help us understand how to parse out the information contained in the planar points relocation time series or walk (recall Eq 12)

$$W = \left\{ \left(t; x_t, y_t\right) \middle| t = 0, \ldots, n^{\text{time}} \right\}$$

From this time series, the following step length $S$, or equivalently velocity $V$ and turning-angle $\Delta\Theta$ time series, can be derived using the relocation point frequency $F$

$$V = \{(t, v_t \mid t = 1, \ldots, n^{\text{time}}\} \tag{A.1}$$

using the derived values

$$v_t = F\sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2} \tag{A.2}$$

and

$$\Delta\Theta = \{(t, \Delta\theta_t \mid t = 2, ..., n^{\text{time}}\} \tag{A.3}$$

using the derived values

$$|\Delta\theta_t| = \min\{|\theta_t - \theta_{t-1}|, 2\pi - |\theta_t - \theta_{t-1}|\}, \qquad \text{where } \theta_t = \arctan\left(\frac{y_t - y_{t-1}}{x_t - x_{t-1}}\right) \tag{A.4}$$

and the sign of $|\Delta\theta_t|$ is easily determined by considering which quadrants $\theta_{t-1}$ and $\theta_t$ are in and whether moving clockwise or anticlockwise subscribes the smaller of the two angles between them: anticlockwise implies positive and clockwise negative. As in [24], we may also study movement in terms of the persistent ($v^{\text{per}}$) and tangential ($v^{\text{tan}}$) velocities rather than speed ($v$ or stepsize) and turning angle ($\delta theta$), using the transformations

$$v_t^{\text{per}} = v_t \cos(\theta_t) \;\text{ and }\; v_t^{\text{tan}} = v_t \sin(\theta_t) \tag{A.5}$$

**FuME** A FuME is a fundamental movement element that for some types may be hard to observe its start and end points, or even define with any precision

**Step** A step is one point $t$ in the time series $W$ with which we associate a step length $s(t)$ and turning angle $\Delta\theta(t)$

**StaME** (previously called metaFuMEs) These are statistically entities computed from consecutive segments of fixed length of a walk $W$ (Eq 12) (i.e., each segment contains a specified number of consecutive relocation points—typically 10-30 to obtain reasonable statistics). These segments can be classified into a few different StaME categories or types based on their statistics—typically the mean and variances of the persistent and turning velocities $v(p)$ and $h_t$ obtained for each segment.

**CAM** A canonical activity mode, formerly referred as a "short" or " homogeneous" CAM that is a consecutive sequence of two or more StaMEs of the same type.

**Activity type** Formerly referred to as a long CAM, and is a characteristic mix of CAMs defining a complex movement activity such as grazing (a mix of stop-start movements)

**DAR** A diel activity routine with start and stop points in its 24 hour cycle selected to correspond, for example, to its major resting period.

**LiMP** A lifetime movement phase is a sequence of DARs that occurs during the individuals life history. It may be a migration event that occurs over several days or weeks, or it may be a seasonal movement pattern (e.g., a nomadic phase, or a home ranging phase) that repeats annually.

**LiT** The lifetime track of an individual from the moment it starts moving after birth to the moment it ceases to move at death.

## A.3 Step-selection procedures

ANIMOVER_1 includes two movement modes, each involving its own step selection procedure. One specifies the movement of individuals within patches (wp) and a second between patches (bp). Each of these two modes involves a kernel with mode-specific parameters $K_\alpha = K\left(r_\alpha^{\min}, r_\alpha^{\max}, \psi_\alpha\right)$, $\alpha =$ wp, bp (Eq 4), as well as mode specific time-in-mode parameter $\hat{t}_\alpha^s$ and neighborhood value parameter $\hat{c}_\alpha^{\text{nbh}}$. The procedures also involve updating individual and cellular array states using Eqs 6 or 7, depending on which of the `RAM`s (see Section 4.2) have been selected.

**Within patches step-selection procedure,** $\mathscr{R}_{\text{wp}}(\hat{t}^{\text{s}}_{\text{wp}}, \hat{c}^{\text{nbh}}_{\text{wp}})$

$\mathscr{R}_{\text{wp}}$.1 *SET* $\alpha_t = \alpha_{\text{wp,next}}$, $(x_t, y_t, \theta_t, t^{\text{s}}_t) = (x_{\text{next}}, y_{\text{next}}, \theta_{\text{next}}, t^{\text{s}}_{\text{next}})$, and $\left(c_{ab,t}, h\right) = \left(c_{ab,\text{next}}, h_{\text{next}}\right)$ (Fig 2).

$\mathscr{R}_{\text{wp}}$.2 *COMPUTE* the values of the admissible set of cells $\mathscr{C}^{\text{wp}}_t$ (Eq 9) and *COMPUTE* $c^{\max}_{ab} = \max\{c_{ab} \,|\, \text{cell}(a,b) \in \mathscr{C}^{\text{wp}}_t\}$. Note, for the case `topology=plane`, there may be no admissible cells.

$\mathscr{R}_{\text{wp}}$.3 *COMPUTE*

$$\text{Nsum}_9\left(c^{\max}_{ab}\right) = c^{\max}_{ab} + \text{ sum of values in 8-cell Moore neighborhood of } c^{\max}_{ab} \text{ cell}$$

$\mathscr{R}_{\text{wp}}$.4 *IF* $\text{Nsum}_9(a,b) \geq \hat{c}^{\text{nbh}}_{\text{wp}}$ *THEN*

(a) *COMPUTE* the set of probabilities $\mathscr{P}^{\text{wp}}_t$ (Eq 10) associated with selecting one of the admissible cells

(b) *SELECT* the next cell, $\text{cell}(a,b)$ using the probabilities in $\mathscr{P}^{\text{wp}}_t$ in a multinomial drawing (i.e., the cell most likely to be selected is the one with the largest $p_{ab}$ and so on)

*ELSE*

(c) *MODIFY* $\tilde{K}^{\text{wp}}$ by replacing $r^{\max}_{\text{wp}}$ with $r^{\max}_{\text{bp}}$, $r^{\min}_{\text{wp}}$ with 0 (typically we already have $r^{\min}_{\text{wp}} = 0$), and $\psi_{\text{wp}}$ with $\pi$ and repeat steps $\mathscr{R}_{wp}$.2 & 3.

(d) *APPLY* rule $\mathscr{R}_{\text{wp}}$ 4a & b to the selected cell.

(e) *IF* the updated $\text{Nsum}_9(a,b) \geq \hat{c}^{\text{nbh}}_{\text{wp}}$ *THEN* move to this cell

(f) *ELSE* move at random to one of the cells on the rim of a circle at radius $r^{\max}_{\text{bp}}$ from the current location (i.e., all points whose distance from the current location is within the band $\left[r^{\max}_{\text{bp}} - \sqrt{\Delta x^2 + \Delta y^2}, r^{\max}_{\text{bp}}\right]$). We note that this step will also prevent an individual from getting stuck at a boundary when `topology = plane`.

$\mathscr{R}_{\text{wp}}$.5 Now that we have identified the next $\text{cell}(a,b)$ to be occupied:

(a) *SET* $(x_{\text{next}}, y_{\text{next}}) = (x_a, y_b)$.

(b) *COMPUTE* the next angle of heading $\theta_{\text{next}}$ (Eq 3) and distance moved $\rho_{ab}(x_{\text{next}}, y_{\text{next}})$ (Eq 2).

(c) *UPDATE* $c_{ab\,\text{next}}$ and $h_{\text{next}}$ using $\texttt{RAM}^{\text{update}}_0$ (Eq 6) or $\texttt{RAM}^{\text{update}}_1$ (Eq 7) or a user specified $\texttt{RAM}^{\text{update}}_2$

$\mathscr{R}_{\text{wp}}$.6 *UPDATE* the StaME index $\alpha_{\text{next}}$ using probability $p_{\text{wp}}(t^{\text{s}})$ (Eq 11) as follows

(where $\text{Nsum}_9(a,b)$ is sum of values in the 8-cell Moore neighborhood of $\text{Cell}(a,b)$ plus $c_{ab}$ itself)

*IF* $Z \sim \texttt{BINOMIAL}[p_{\text{wp}}(t^{\text{s}})] = 1$ *AND* $\text{Nsum}_9(a,b) \geq \hat{c}^{\text{nbh}}_{\text{wp}}$ (if updated under $\mathscr{R}$.4c use latest value) *THEN* $\alpha_{\text{next}} = \text{wp}$ *ELSE* $\alpha_{\text{next}} = \text{bp}$

**Between patches step-selection procedure,** $\mathscr{R}_{\text{bp}}(\hat{t}^{\text{s}}_{\text{bp}}, \hat{c}^{\text{nbh}}_{\text{bp}})$

$\mathscr{R}_{\text{bp}}$.1 *SET* $\alpha_t = \alpha_{\text{bp,next}}$, $(x_t, y_t, \theta_t, t^{\text{s}}_t) = (x_{\text{next}}, y_{\text{next}}, \theta_{\text{next}}, t^{\text{s}}_{\text{next}})$, and $\left(c_{ab,t}, h\right) = \left(c_{ab,\text{next}}, h_{\text{next}}\right)$ (Fig 2).

$\mathscr{R}_{\text{bp}}$.2 *COMPUTE* the values of the admissible set of cells $\mathscr{C}^{\text{vis}}_t$ (i.e., using $\tilde{K}^{\text{vis}}_t(0, r^{\max}_{\text{wp}}, \pi/2)$ rather than $\tilde{K}^{\text{bp}}_t$ in Eq 9) and *COMPUTE* $c^{\max}_{ab} = \max\{c_{ab} \,|\, \text{cell}(a,b) \in \mathscr{C}^{\text{vis}}_t\}$. Note, for the case `topology=plane`, there may be no admissible cells when the individual is on the boundary near a corner. In this case, when no visible cells are available, then use $K_{\text{vis}}(0, r^{\max}_{\text{bp}}, \pi)$, i.e., the individual now looks behind itself as well.

$\mathscr{R}_{\text{bp}}$.3 *COMPUTE*

$$\text{Nsum}_9\left(c^{\max}_{ab}\right) = c^{\max}_{ab} + \text{ sum of values in 8-cell Moore neighborhood of } c^{\max}_{ab} \text{ cell}$$

$\mathscr{R}_{\text{bp}}$.4 *IF* $\text{Nsum}_9(a,b) \geq \hat{c}^{\text{nbh}}_{\text{wp}}$ *THEN*

(a) *COMPUTE* the set of probabilities $\mathscr{P}^{\text{wp}}_t$ (Eq 10) associated with selecting one of the admissible cells

(b) *SELECT* the next cell, $\text{cell}(a,b)$ using the probabilities in in a multinomial drawing

*ELSE* move to any cell at random in the set $\mathscr{C}^{\text{bp}}_t$ (i.e., overlapping with $\tilde{K}^{\text{wp}}_t$).

$\mathscr{R}_{\text{bp}}$.5 Now that we have identified the next $\text{cell}(a,b)$ to be occupied:

(a) *SET* $(x_{\text{next}}, y_{\text{next}}) = (x_a, y_b)$.

(b) *COMPUTE* the next angle of heading $\theta_{\text{next}}$ (Eq 3) and distance moved $\rho_{ab}(x_{\text{next}}, y_{\text{next}})$ (Eq 2).

(c) *UPDATE* $c_{ab\,\text{next}}$ and $h_{\text{next}}$, but for this movement mode setting $\kappa^{\text{add}} = 0$ in the same $\texttt{RAM}^{\text{update}}$ used in $\mathscr{R}_{\text{wp}}$.6

$\mathscr{R}_{\text{bp}}$.6 *UPDATE* the StaME index $\alpha_{\text{next}}$ using probability $p_{\text{bp}}(t^{\text{s}})$ (Eq 11) as follows

(where $\text{Nsum}_9(a,b)$ is sum of values in the 8-cell Moore neighborhood of $\text{Cell}(a,b)$ plus $c_{ab}$ itself)

*IF* $Z \sim \texttt{BINOMIAL}[p_{\text{bp}}(t^{\text{s}})] = 1$ *AND* $\text{Nsum}_9(a,b) \leq \hat{c}^{\text{nbh}}_{\text{bp}}$ *THEN* $\alpha_{\text{next}} = \text{bp}$ *ELSE* $\alpha_{\text{next}} = \text{wp}$

## A.4 StaMEs from empirical data

Various methods can be applied to extract StaMEs from empirical data. We have outlined one method in Section 3 of our text. Here we present more comprehensive details on the method we use. In order to separate different movement modes from each other, we perform a cluster analysis. Clustering is an unsupervised machine learning procedure which finds an optimum partition of a set of points with two or more variables into subsets (or clusters) of similar points.

Here we use hierarchical clustering approach, which results in a tree structure called dendrogram having a single cluster at the root, with leaf nodes representing data points in the set. Recall from Eq 13 that the variables $\mathscr{S}_\mu$ had been constructed using the relocation time series $W$ in Eq 12. which we repeat here for convenience:

$$\mathscr{S}_\mu = \left\{ \text{Seg}_z\big(V_z, \text{SD}^V_z, |\Delta\Theta|_z, \text{SD}^{|\Delta\Theta|}_z, \Delta^\rho\big) \big| z = \lfloor \frac{t}{\mu} \rfloor + 1,\ t = 0, \cdots, n^{\text{time}} - 1 \right\} \text{ such that}$$

$$V_z = \frac{\sum_{\tau=t}^{t+\mu-1} v_\tau}{v_{\max}\mu} \text{ with st. dev. } \text{SD}^V_z,\ |\Delta\Theta|_z = \frac{\sum_{\tau=t}^{t+\mu-1} |\Delta\theta_\tau|}{2\pi\mu} \text{ with st. dev. } \text{SD}^{|\Delta\Theta|}_z$$

$$\text{and } \Delta^\rho = \frac{\sqrt{(x_{t+\mu} - x_t)^2 + (y_{t+\mu} - y_t)^2}}{\mu V_z}.$$

To begin with, it is important to discard any points containing missing or non-numeric values (since the variables to be used in clustering algorithm are all numeric). Further, each of these variables ranges in $[0,1]$, and hence no further normalisation has been carried out. Cluster analysis is performed on the segment level data (i.e., with $n_{\text{seg}} = \lfloor \frac{t}{\mu} \rfloor + 1$ points) using the variables in $\mathscr{S}_\mu$. Specifically, we want to find an optimal partition

$$\mathscr{P}(\mathscr{S}_\mu) = \{C_1, C_2, ..., C_k\},$$

where $\cup_{i=1}^k C_i = \mathscr{S}_\mu$, and $C_i \cap C_j = \emptyset$ (hard clustering).

Hierarchical agglomerative algorithms implement bottom-up clustering methodology, which starts with each point of $\mathscr{S}$ in its own singleton cluster, followed by successive fusions of two clusters at a time, depending on similarity between them, leading to a specified number of clusters (or, alternatively, leading to one cluster followed by cutting the dendrogram at the desired number of clusters). These are deterministic, yet greedy, in the sense that the clusters are merged based entirely on the similarity measure, thereby yielding a local solution. Most of the similarity schemes are specified not in terms of an objective function to be optimized, but procedurally. We use Ward's minimum sum of

square scheme, which performs a fusion of clusters while minimizing the intra-cluster variance. Our distance metric to quantify dissimilarity is the Euclidean measure.

To perform clustering, we make use of `hclust` function from `fastcluster` `R` package. It replaces the `stats::hclust` function, which offers the most common implementation of Ward's hierarchical clustering in `R`. The conventional algorithm from `stats` package takes as input the set of points $\{\mathbf{x}_z\}|_{z=1}^{n_{\text{seg}}} = \left\{V_z,\ \text{SD}_z^V,\ |\Delta\Theta|_z,\ \text{SD}_z^{|\Delta\Theta|}, \Delta_z^{\rho}\right\}\Big|_{z=1}^{n_{\text{seg}}} \in \mathbb{R}^{n_{\text{seg}}\times 5}$ (represented by $\mathscr{S}_\mu$) and a pair-wise dissimilarity matrix $d(C_i, C_j)|_{i,j=1}^{n_{\text{seg}}}$ (to be computed in advance). Starting with $n_{\text{seg}}$ clusters, it fuses multiple pairs of them in each of $n_{\text{seg}} - 1$ steps while satisfying Ward's criterion and updates the dissimilarity matrix with the distance of each of the clusters still available to be merged with the newly created cluster. Accordingly, a series of partitions are produced, with the first one containing singleton clusters and the last one containing all $n_{\text{seg}}$ points in one cluster. At each step, a set of clusters available to be merged (active clusters) is maintained, and each merger is tracked leading to a dendrogram. Concisely, the algorithm can be represented as shown in Algorithm 1. The dendrogram thus produced can also be understood as a weighted graph, with leaf nodes

**Algorithm 1:** Naive hierarchical agglomerative clustering algorithm

```
Initialise active set with each instance in singleton cluster
repeat
    Select the pair of clusters with the most suitable distance
    Remove each from the active set
    Add their union to the tree
until Active set has only one item
```

representing data points, and each internal node representing the cluster of its descendent leaves. The dissimilarity between clusters is represented by edge weights. Built-in `R` function `cutree` can then be used to cut the dendrogram at the required number of clusters, and `colMeans` can be used to compute a cluster's center.

This algorithm has a time complexity of $\mathscr{O}(n_{\text{seg}}^3)$ and requires $\Omega(n_{\text{seg}}^2)$ memory (on account of distance matrix computation and storage), making it unsuitable for our segment-level data set with $\sim\ 10^5$ points. Further, it is also difficult to distribute over multiple threads because the complete dissimilarity matrix along with active clusters and current state of dendrogram is required by all the processes. To get around these difficulties, we make two modifications to this workflow. First, we make use of `parDist` function of `parallelDist` package, which permits parallel computations of pair-wise dissimilarities. It offers the same interface as that of `stats::dist` `R` function. Second, we use an improved algorithm for hierarchical agglomerative clustering from `fastcluster` package, as mentioned above. The algorithm performs hierarchical clustering with Ward's scheme faster by accomplishing the search for the best cluster to merge with any cluster in the most efficient way [80]. For the clustering procedure, we choose to not perform dimensional reduction and use all 5 variables in $\mathscr{S}_\mu$. This ensures enhanced interpretation of results.

As noted above, the number of clusters $k$ needs to be specified in order to cut the dendrogram. As is evident from Figures 6 and 7, we selected $k = 8$ to be used in our analyses. We initially choose to err on the side of having more than fewer clusters than may turn out to be optimal because we wanted to try to isolate clusters at the extremes of the cluster groups that were as homogeneous as possible in terms of not mixing many different movement modes in one movement segment. It may also be useful in some particular cases to use a gap statistic to justify one's choice of $k$. For example, in the two-movement mode simulation with 10-point segmentation we used a Gap statistic that compares the total intra-cluster variance for different values of $k$ with the corresponding no clustering case. In this case, we see in Fig A.1 that the rate of increase of our Gap statistic somewhat flattens at $k = 8$. Accordingly, this would be the appropriate choice for our 10-point segmentation two-mode movement simulation. A relatively large value large value of $k$ allows us to consider how close the centroids of different clusters would be and what it may mean to combine one or more clusters. Also, every data set may have a different preferential value for $k$ when considering a gap statistic. Further, to facilitate comparisons across our analyses of different data sets, we kept to our selection of $k = 8$. As mentioned in the discussion section, hierarchical clustering is but one of many methods that may be used to cluster the data and other machine learning and even deep learning methods [81, 82, 83], can be explored, particularly when segment shapes vary widely.

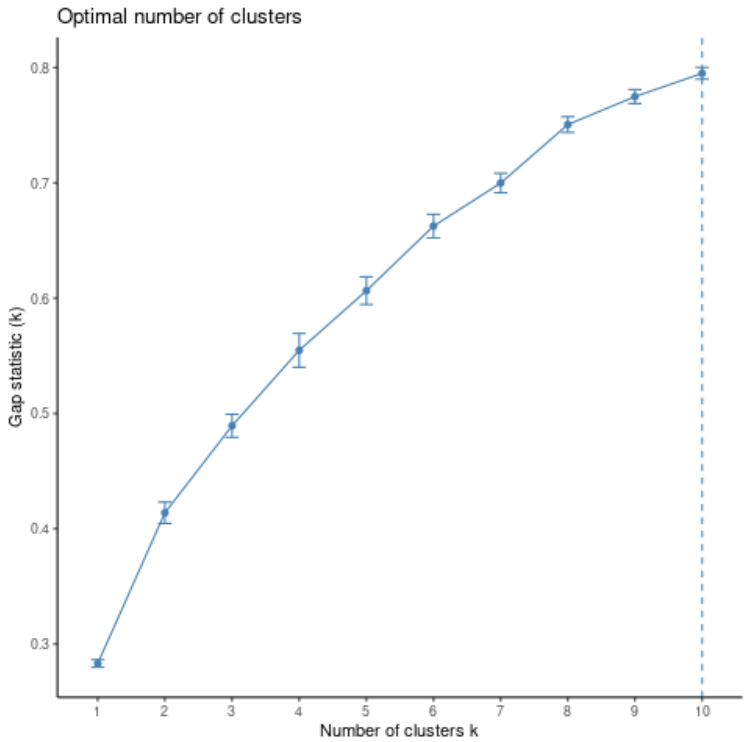


Figure A.1: Gap statistic for the two-movement mode simulation data in section 5.2.

Our computations have been performed on a node of Berkeley's HPC Cluster Savio. The system is Lenovo NeXtScale nx360m5 equipped with two Intel Xeon 12-core Haswell processors (core frequency 2.3 Ghz) and 128 GB of 2133 Mhz DDR4 memory. The `R` function `parDist` distributes distance matrix calculation over all of 24 available processor cores, and returns a "`dist`" class object - the lower triangle of the distance matrix (since it is symmetric). `hclust` returns a list with several components describing the dendrogram. `cutree` method, which is used to cut the tree, returns a vector of integers indicating which cluster each point has been assigned to.

The segment level data in $\mathscr{S}$ has 5 variables. Accordingly, principal components analysis is required to visualize the clustering results. This is a coordinate transformation procedure to a new set of variables such that maximum variance is preserved (which amounts to minimum loss of information). The two principal components accounting for maximum variance are used for 2-d cluster plots.

Our speed and turning-angle measurements are scaled with maximum speed and $\pi$, respectively, ensuring that the segment level data has all variables ranging in $[0,1]$. This makes it reasonable to calculate principal components without further normalisation. For better interpretation, we report the variable $\overline{V} \pm \overline{\mathrm{SD}^V}$ in Table 2 in units of $m/s$ after rescaling with the maximum speed. PCA has been performed using built-in R method `prcomp`. To ensure no further normalisation, the binary arguments `center` and `scale` in `prcomp` (which govern whether the variables are zero centered, and whether the variables are scaled to have unit variance, respectively) are set to `FALSE`. The function returns a list with various components which can be used to read eigenvectors and eigenvalues (and hence standard deviations of principal components) of the covariance matrix.

## A.5 Plots of 10-point simulation data segments

Each plot in this subsection, particularly across different clusters, has had its axes for each panel automatically set by the plotting routine. Thus the size of segments is not comparable, only their relative shapes is informative.

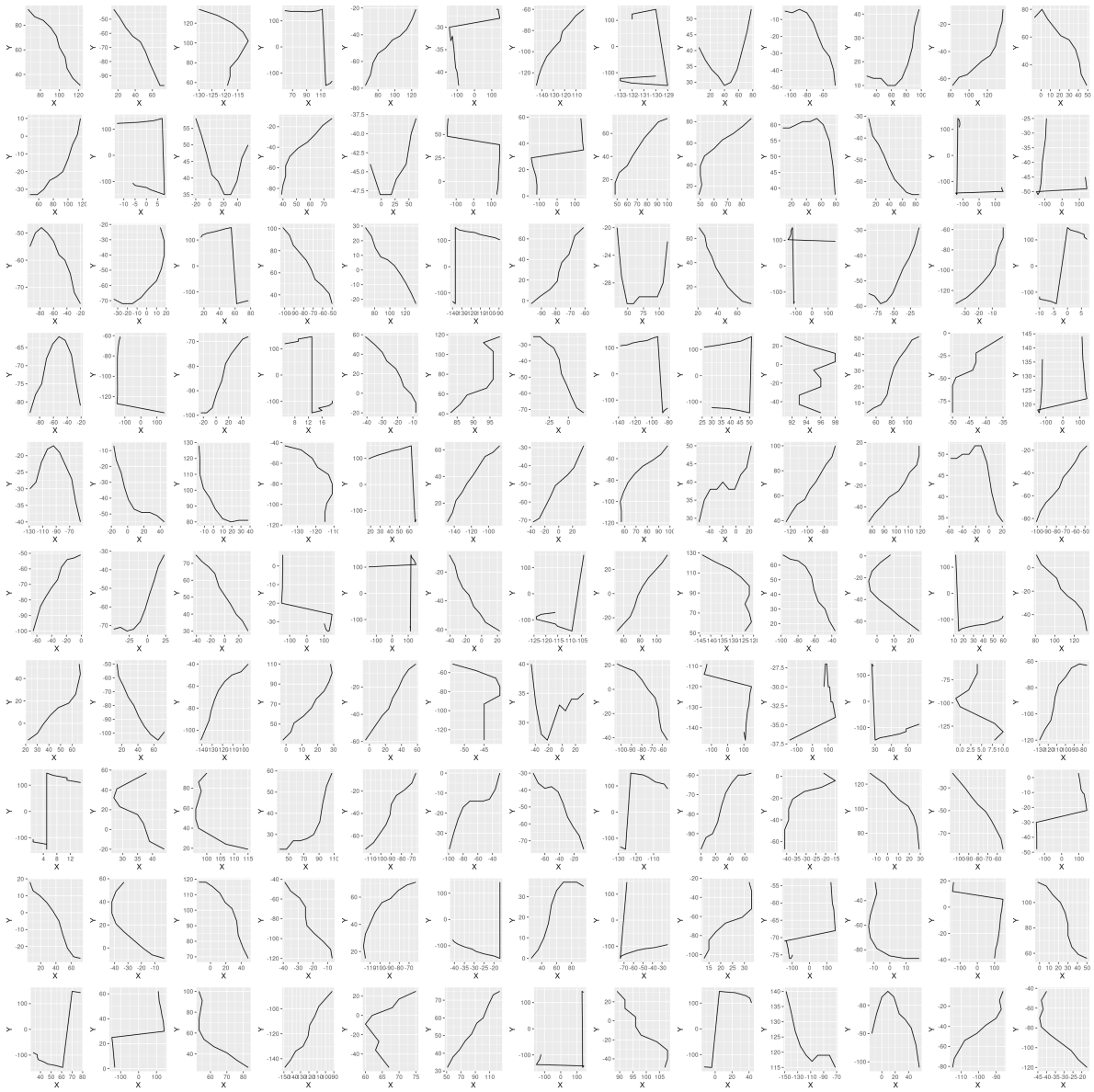

Figure A.2: 130 random 10-point segments from cluster C=1 (Table 1)

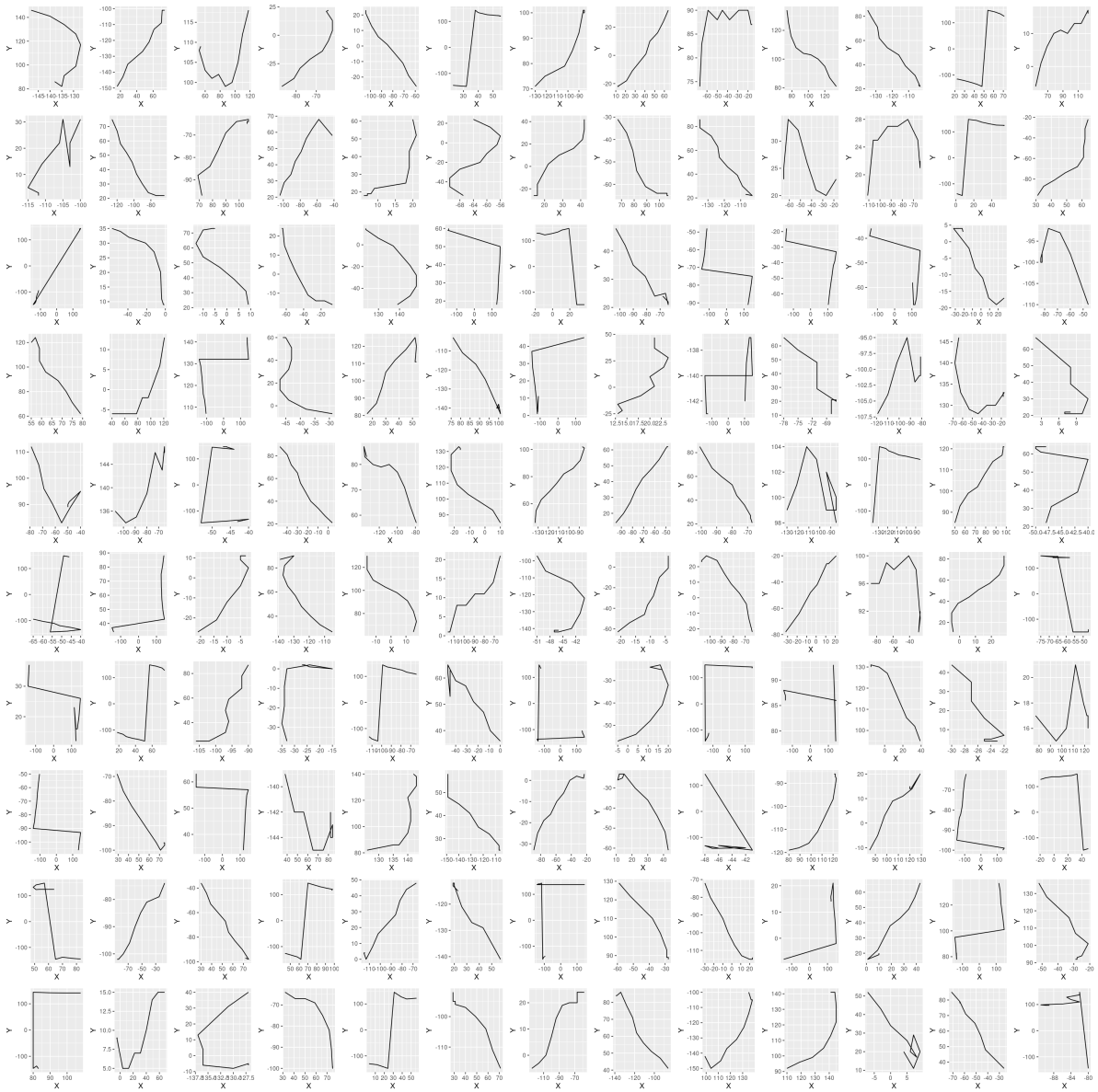

Figure A.3: 130 random 10-point segments from cluster C=2 (Table 1)

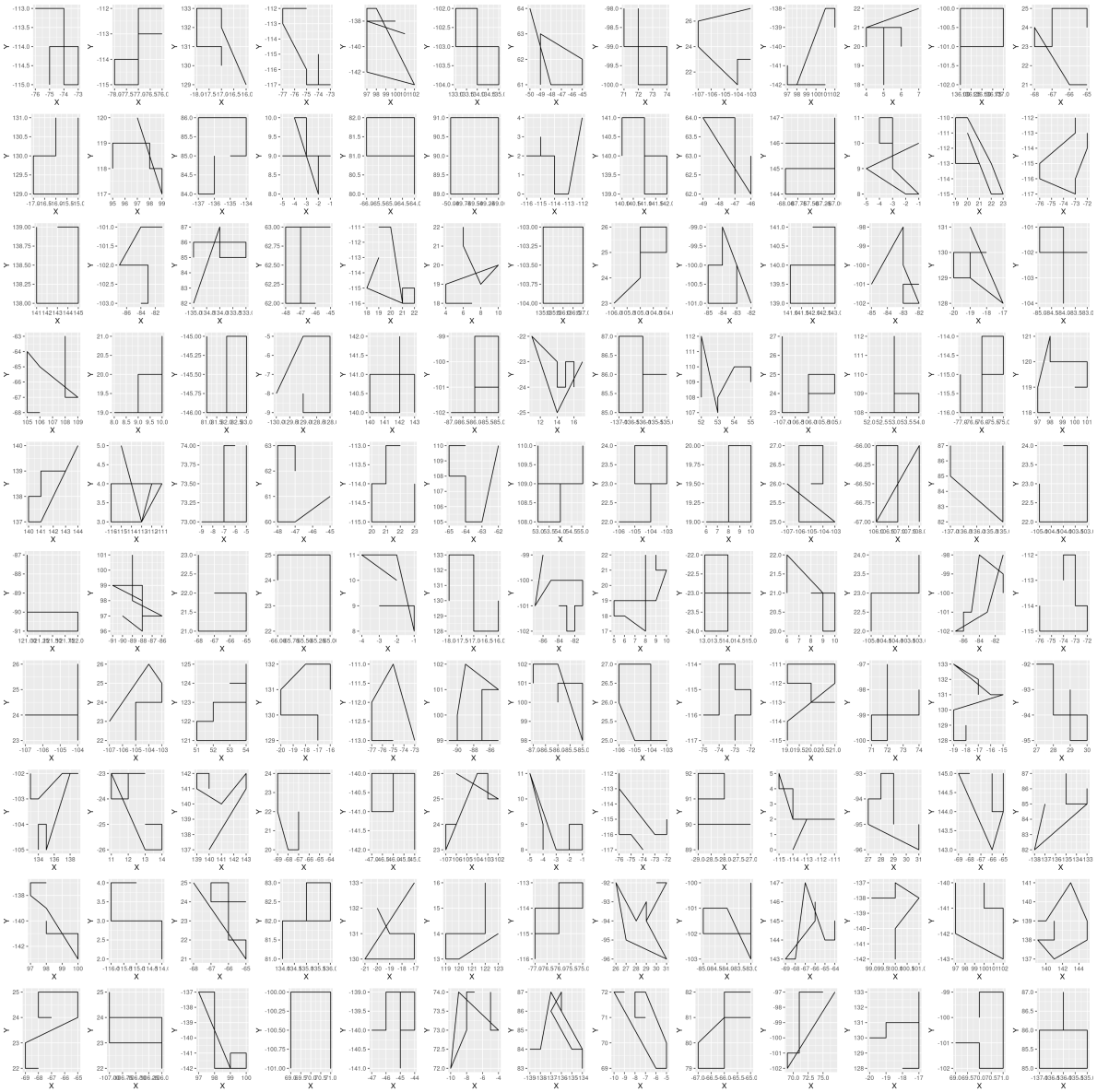

Figure A.4: 130 random 10-point segments from cluster C=3 (Table 1)

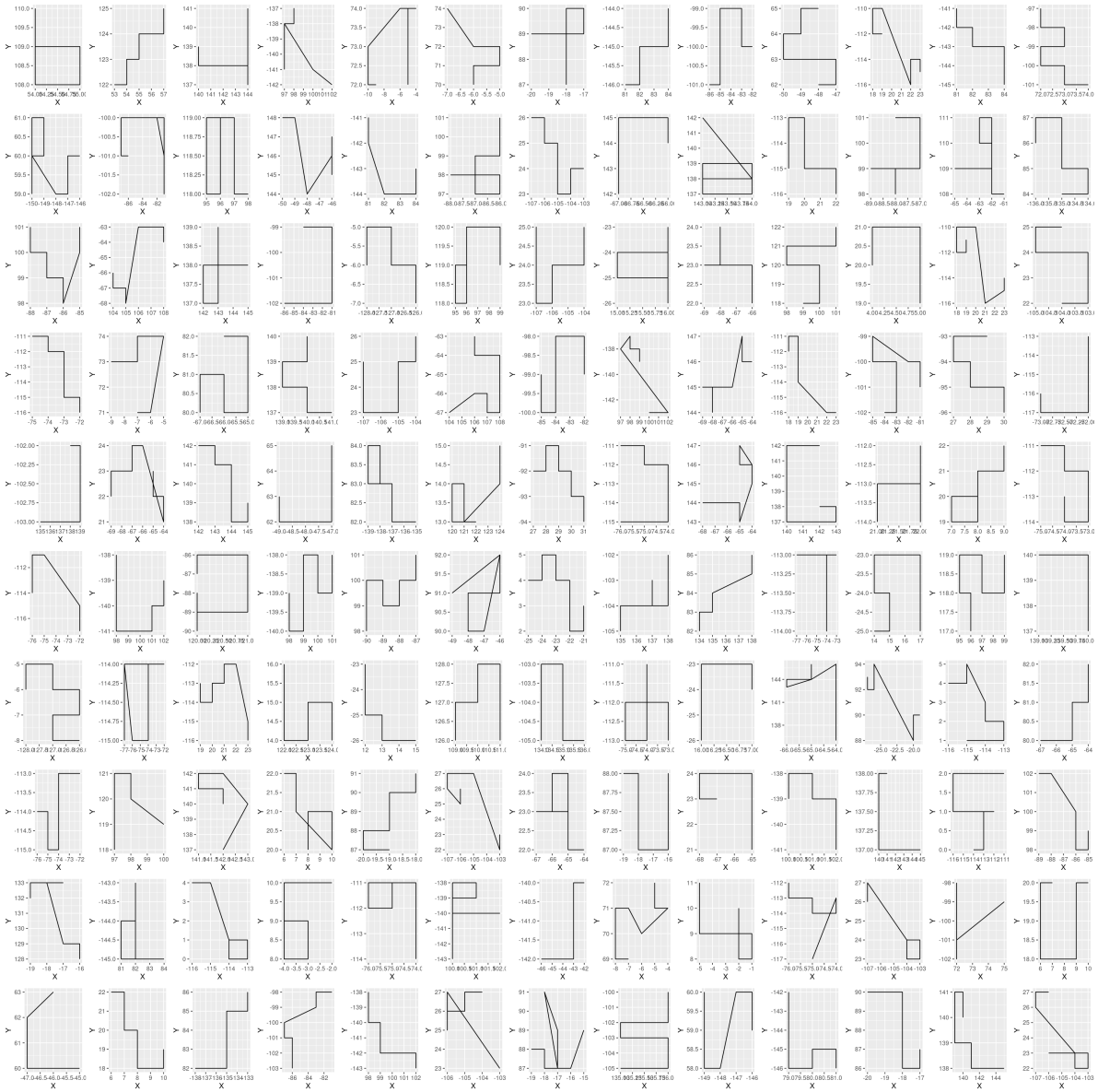

Figure A.5: 130 random 10-point segments from cluster C=4 (Table 1)

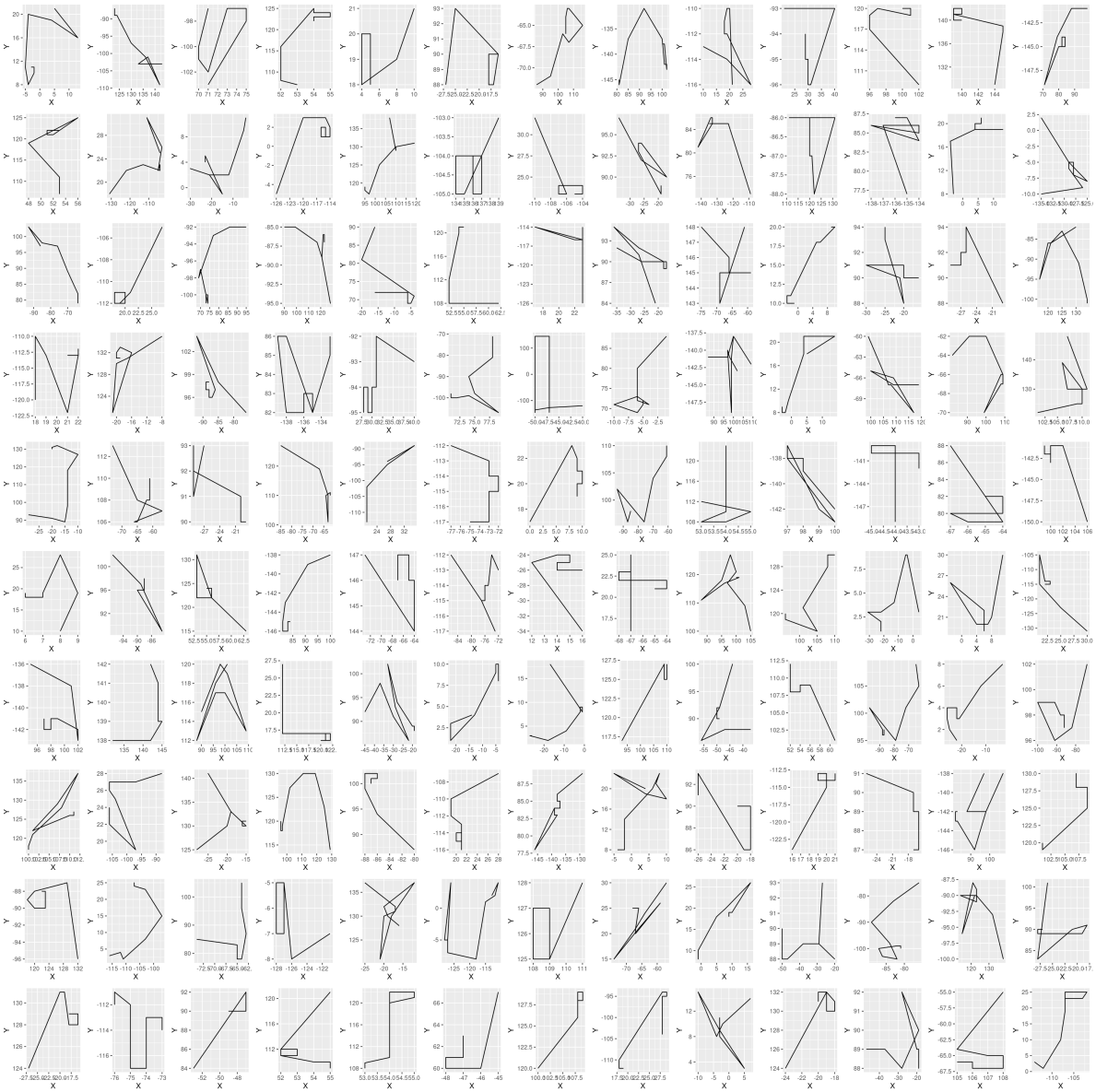

Figure A.6: 130 random 10-point segments from cluster C=5 (Table 1)

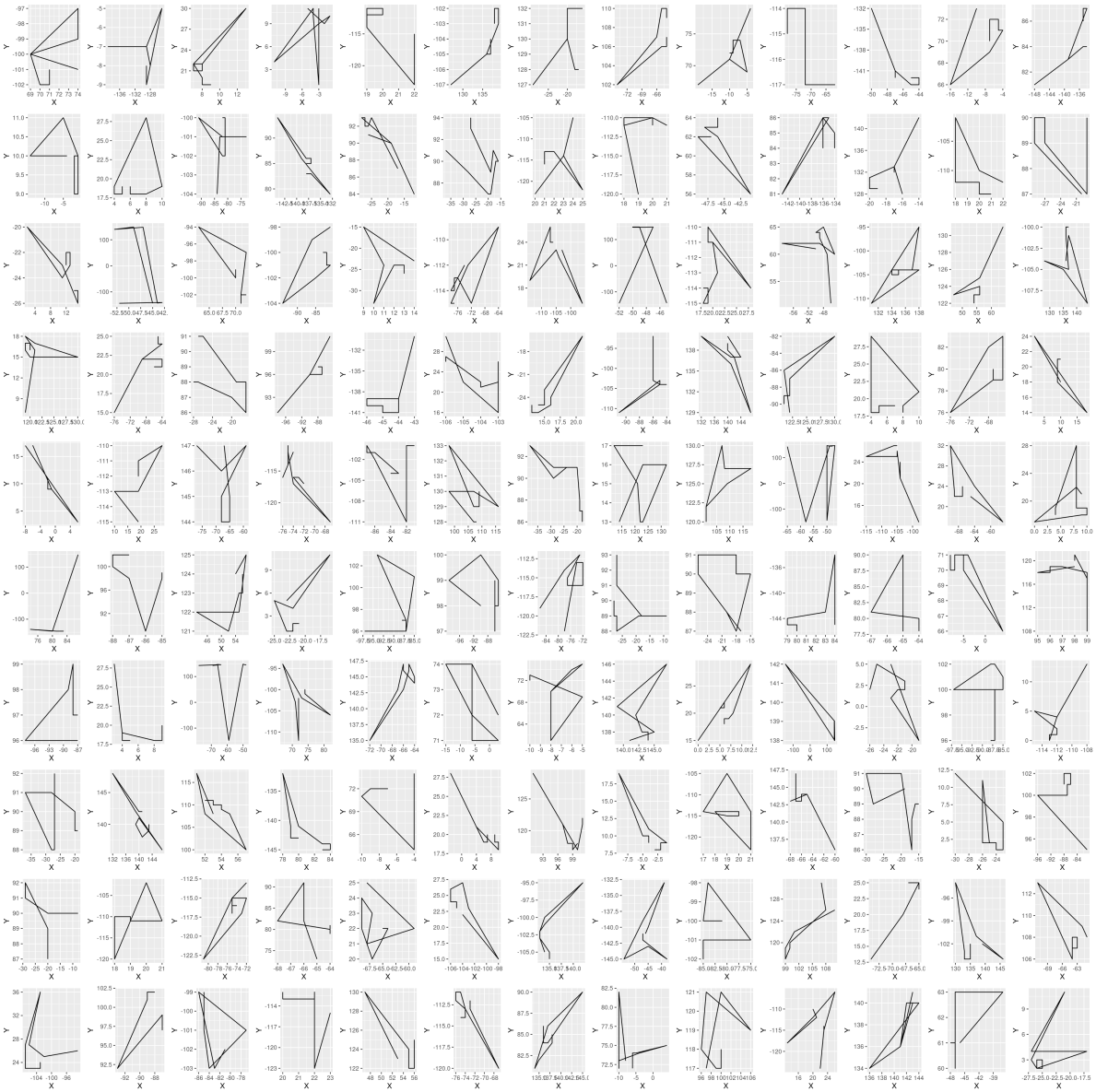

Figure A.7: 130 random 10-point segments from cluster C=6 (Table 1)

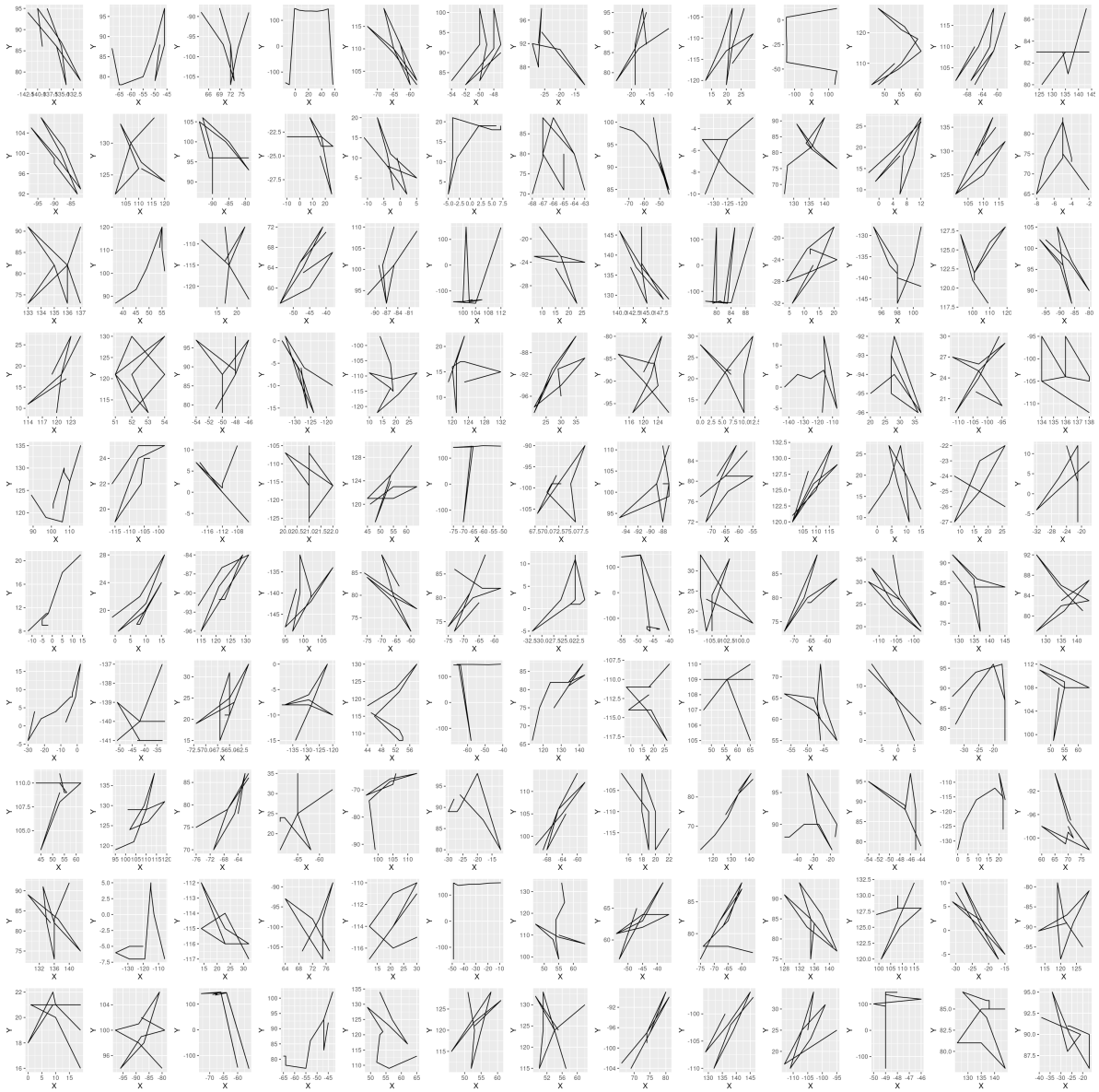

Figure A.8: 130 random 10-point segments from cluster C=7 (Table 1)

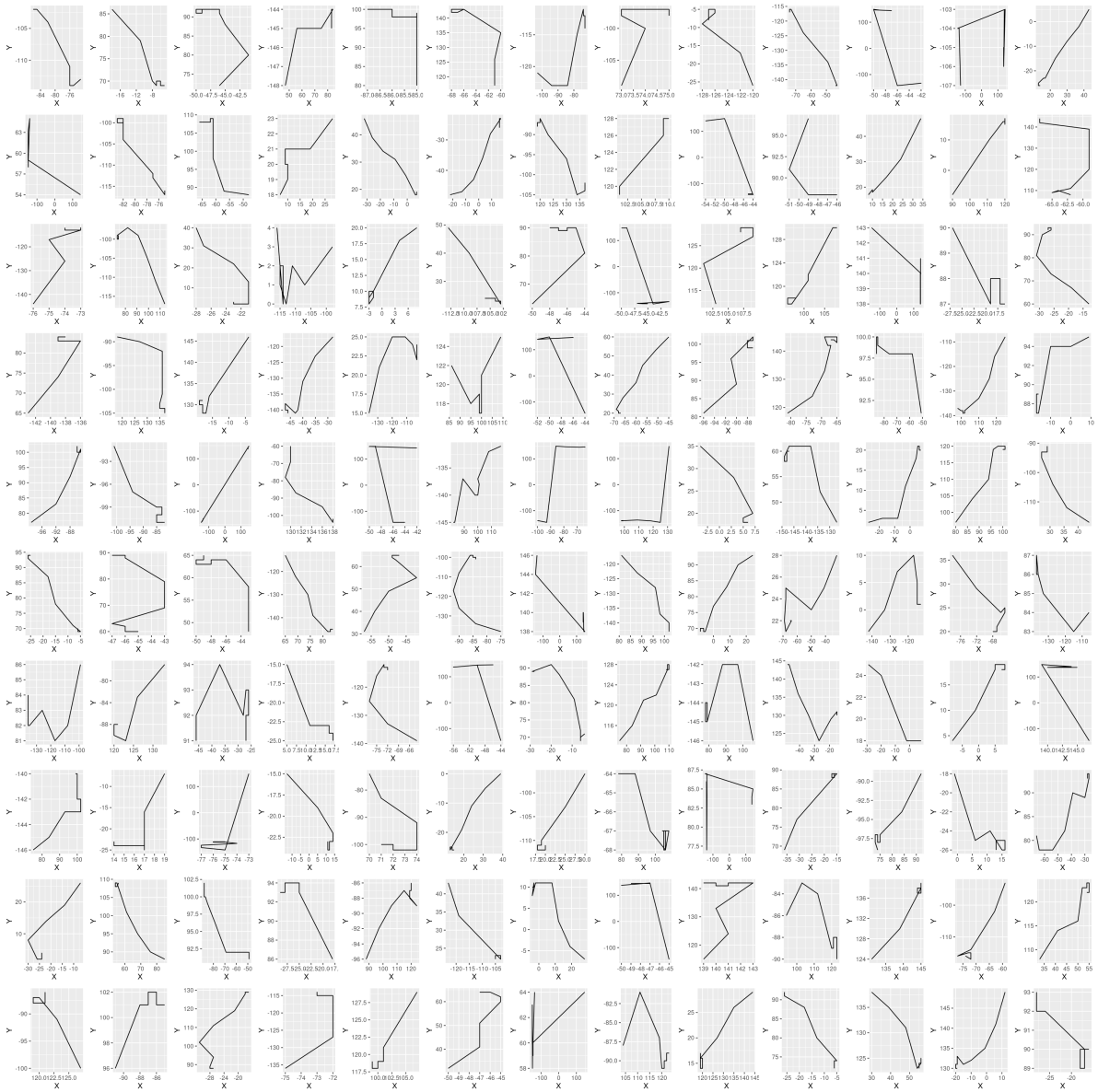

Figure A.9: 130 random 10-point segments from cluster C=8 (Table 1)

## A.6 Plots of Adult Female Barn Owl StaMEs

Each plot in this subsection, particularly across different clusters, has had its axes for each panel automatically set by the plotting routine. Thus the size of segments is not comparable, only their relative shapes is informative.

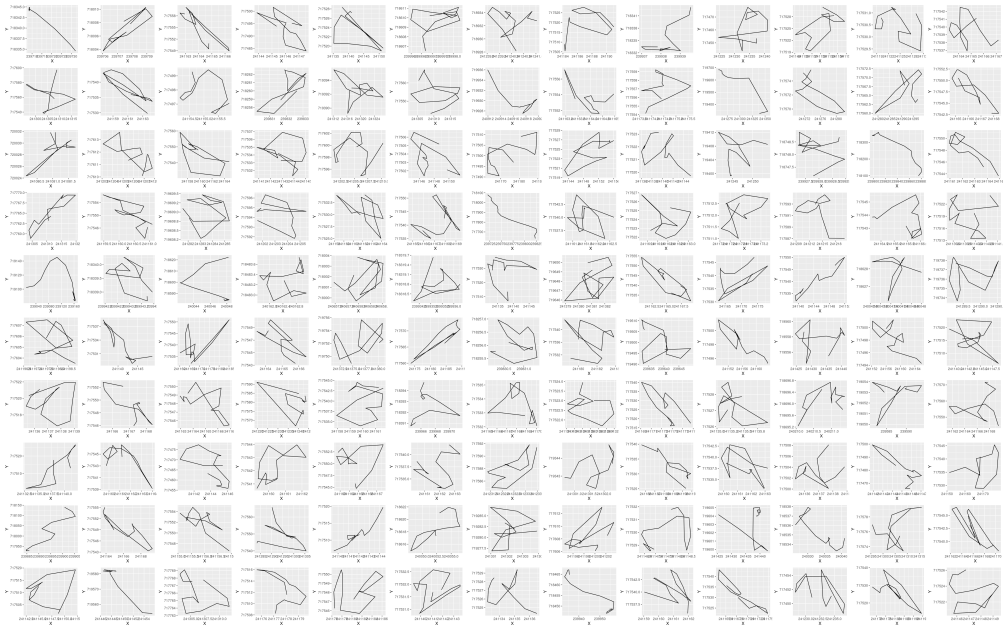

Figure A.10: 130 random adult female barn owl segments from cluster C=1 (Table 2)

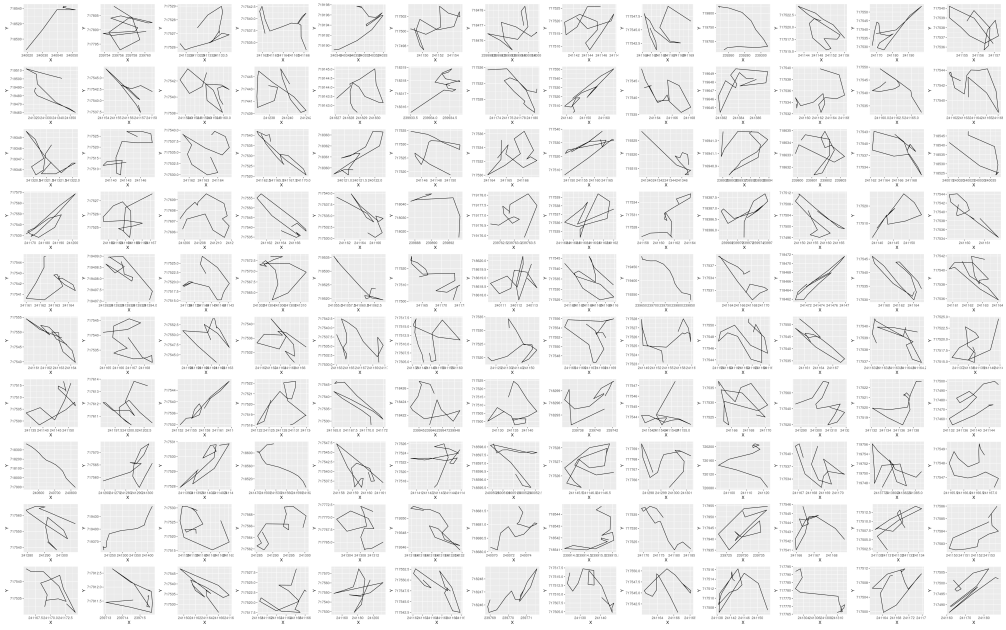

Figure A.11: 130 random adult female barn owl segments from cluster C=2 (Table 2)

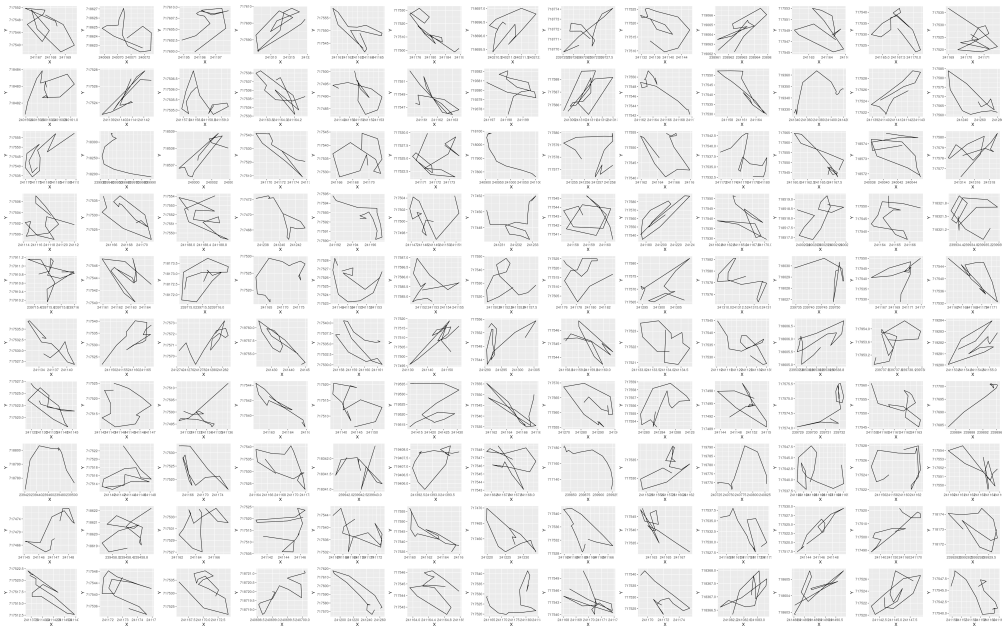

Figure A.12: 130 random adult female barn owl segments from cluster C=3 (Table 2)

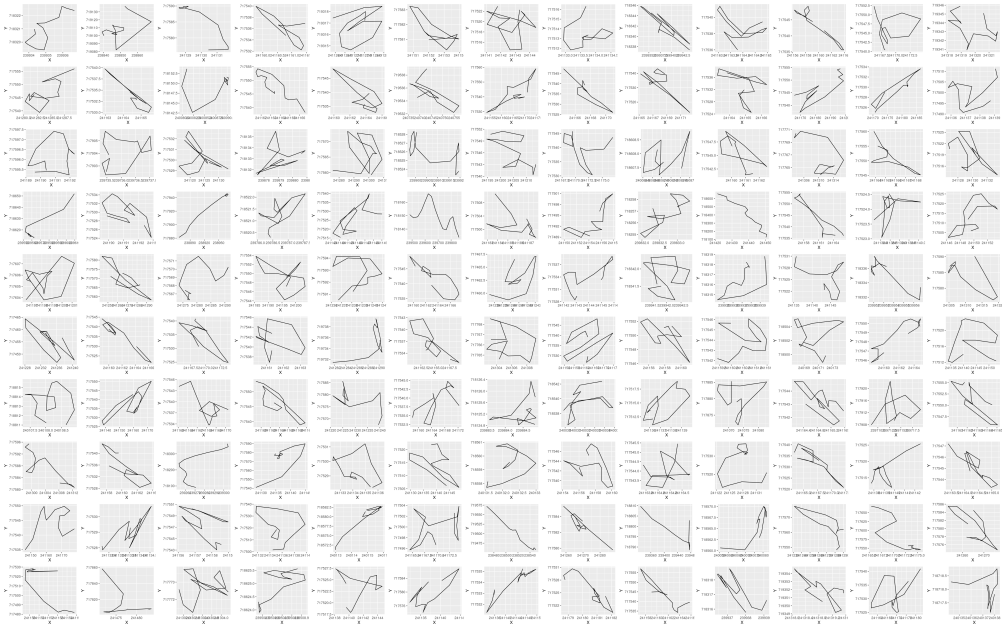

Figure A.13: 130 random adult female barn owl segments from cluster C=4 (Table 2)

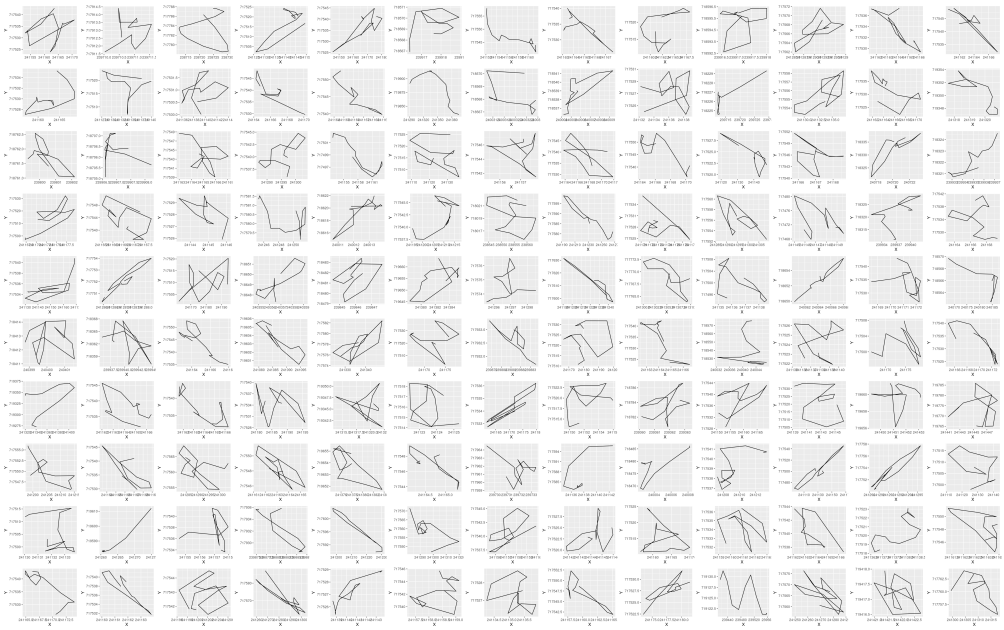

Figure A.14: 130 random adult female barn owl segments from cluster C=5 (Table 2)

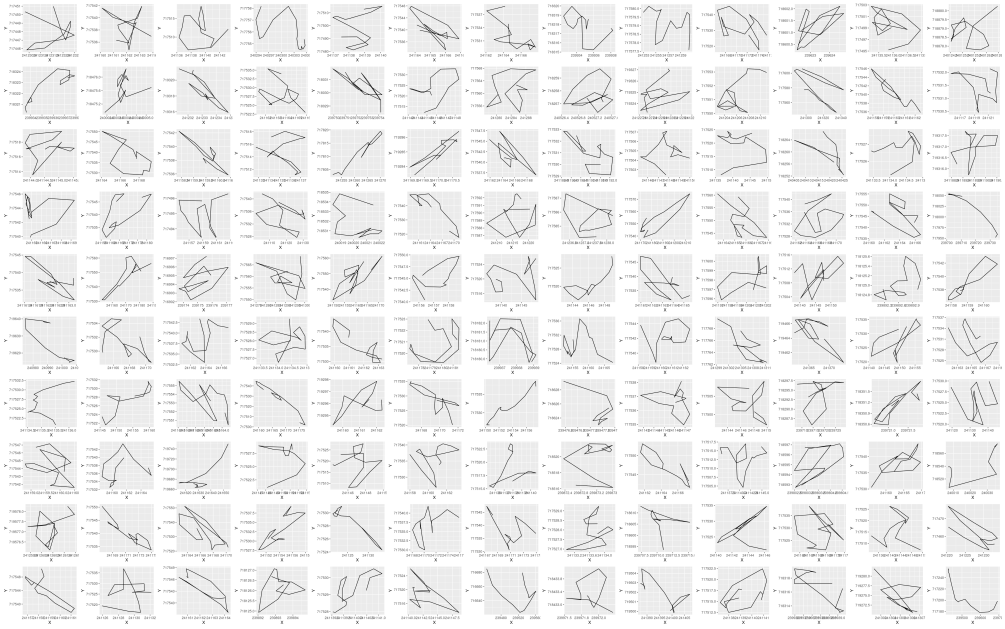

Figure A.15: 130 random adult female barn owl segments from cluster C=6 (Table 2)

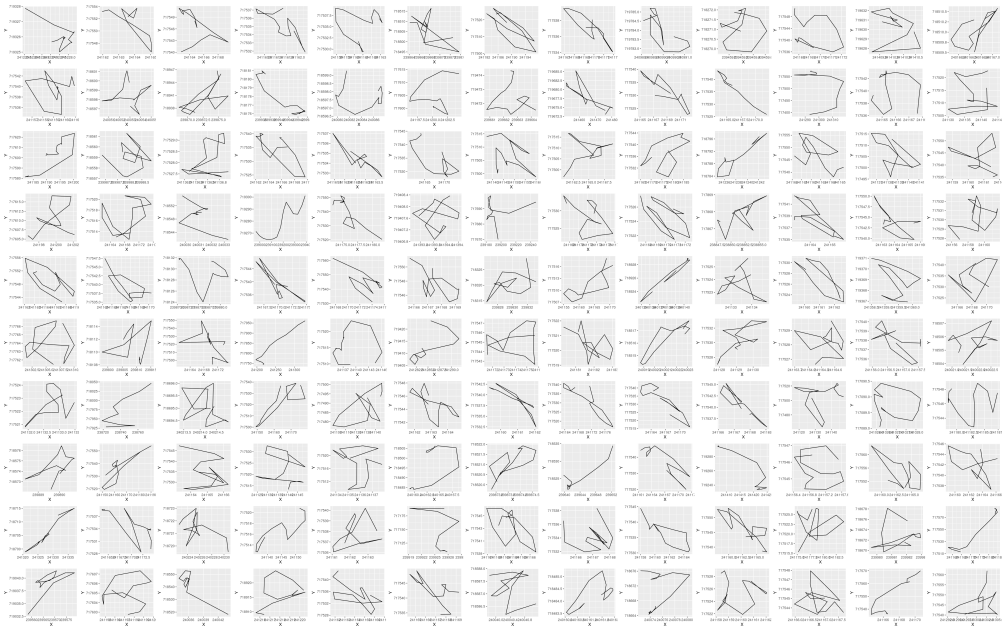

Figure A.16: 130 random adult female barn owl segments from cluster C=7 (Table 2)

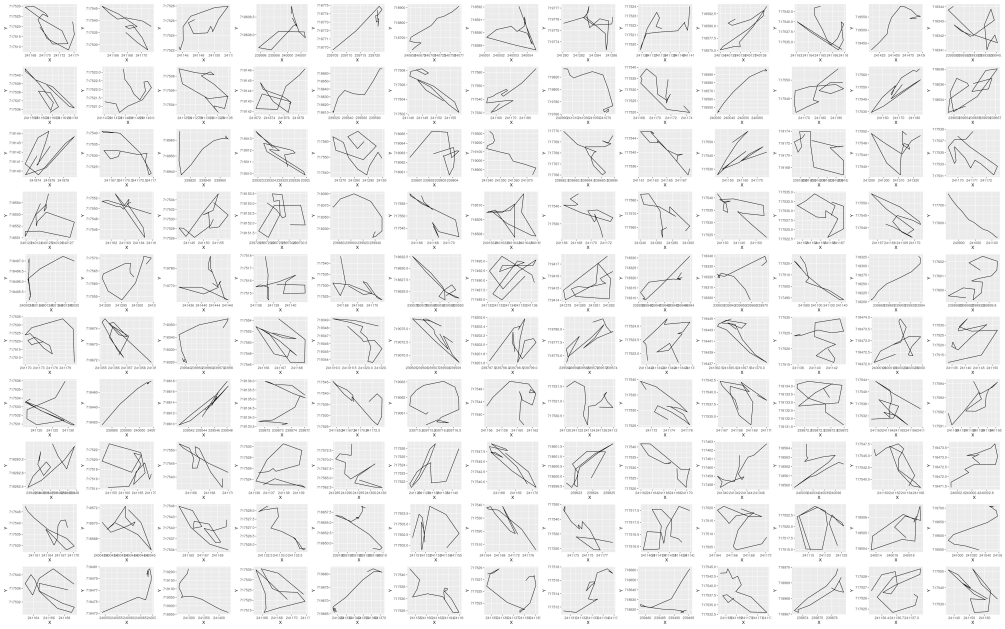

Figure A.17: 130 random adult female barn owl segments from cluster C=8 (Table 2)

## A.7 Plots of Juvenile Male Barn Owl StaMEs

Each plot in this subsection, particularly across different clusters, has had its axes for each panel automatically set by the plotting routine. Thus the size of segments is not comparable, only their relative shapes is informative.

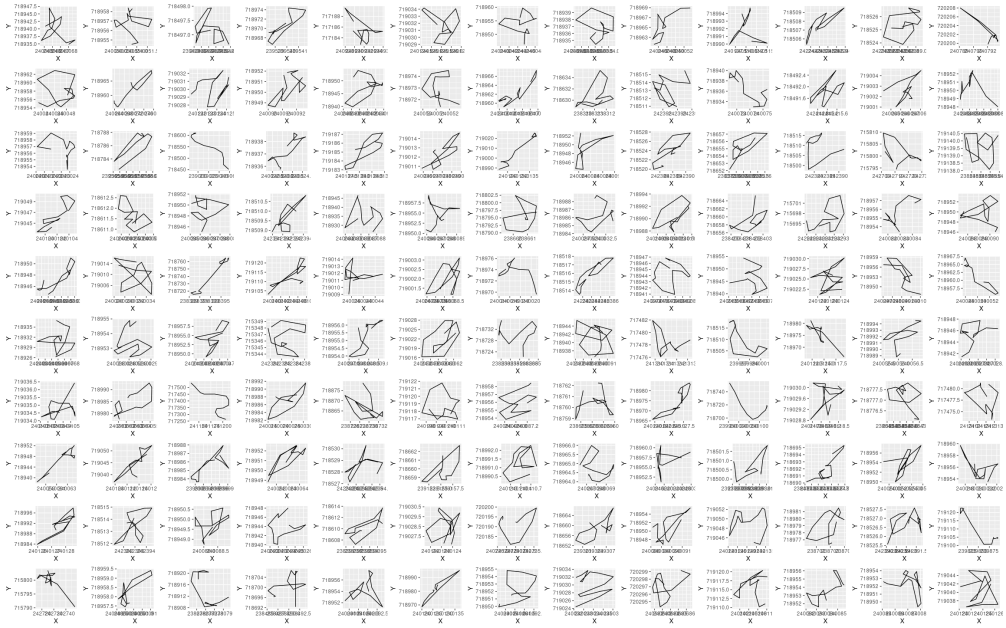

Figure A.18: 130 random adult female barn owl segments from cluster C=1 (Table 2)

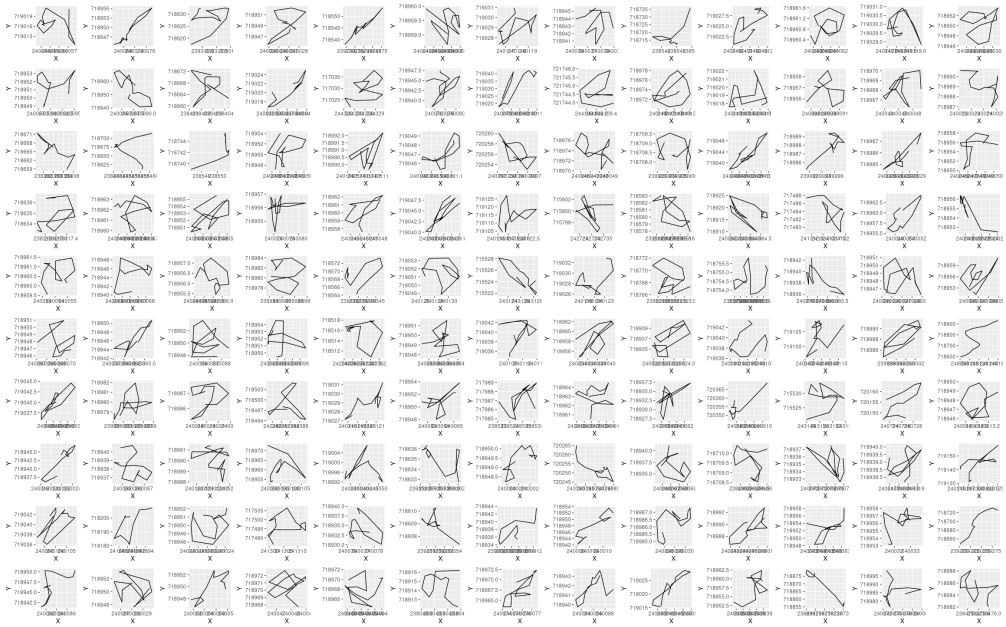

Figure A.19: 130 random juvenile barn owl segments from cluster C=2 (Table 2)

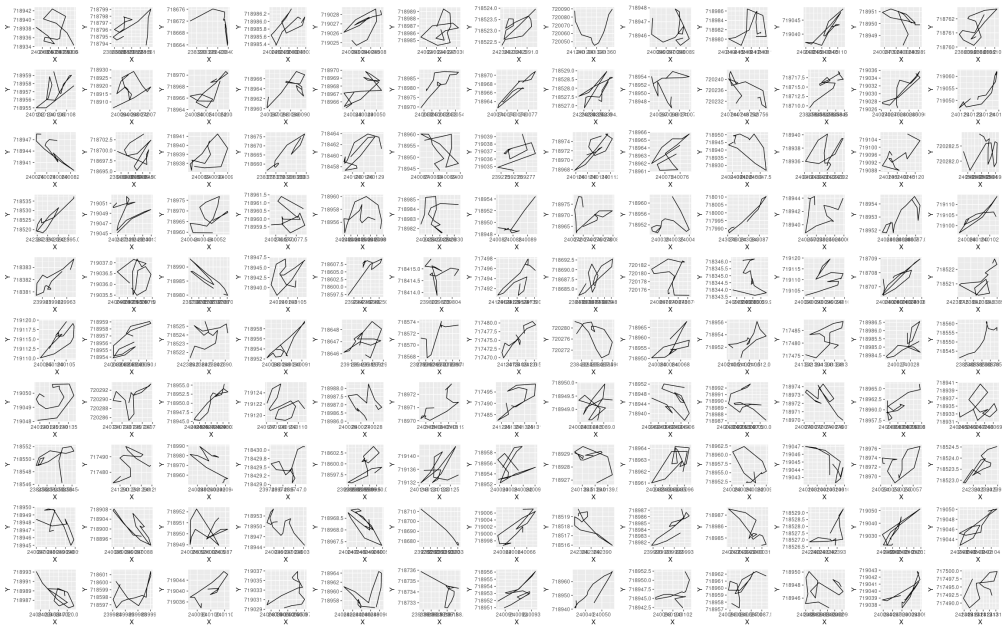

Figure A.20: 130 random juvenile barn owl segments from cluster C=3 (Table 2)

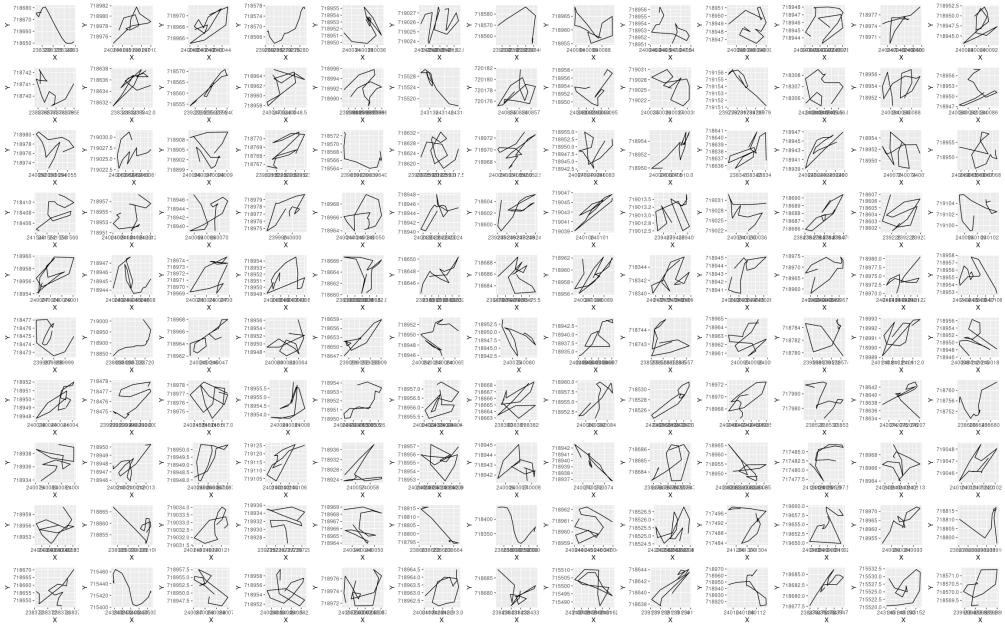

Figure A.21: 130 random juvenile barn owl segments from cluster C=4 (Table 2)

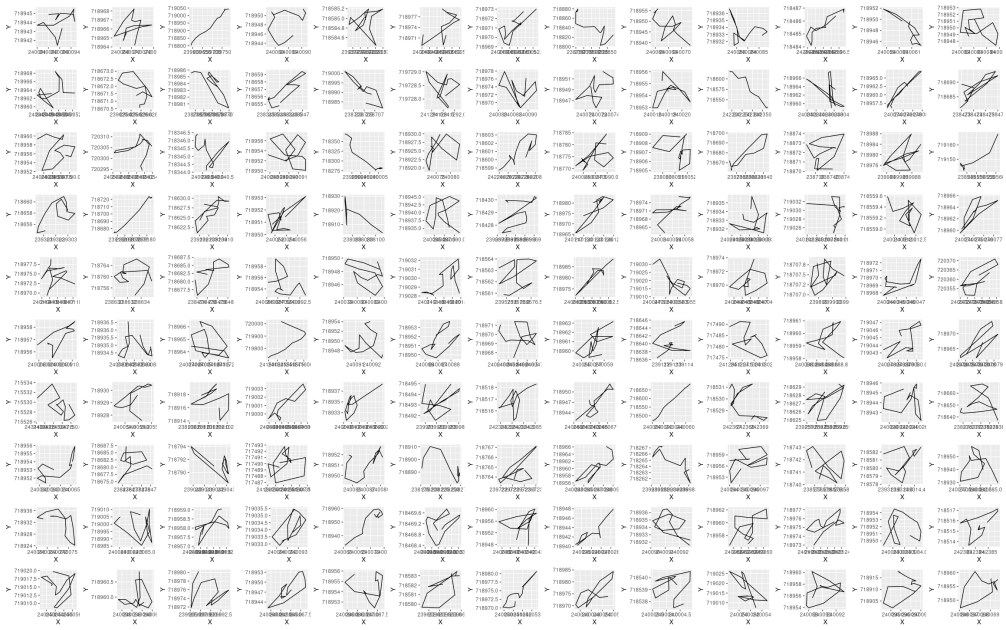

Figure A.22: 130 random juvenile barn owl segments from cluster C=5 (Table 2)

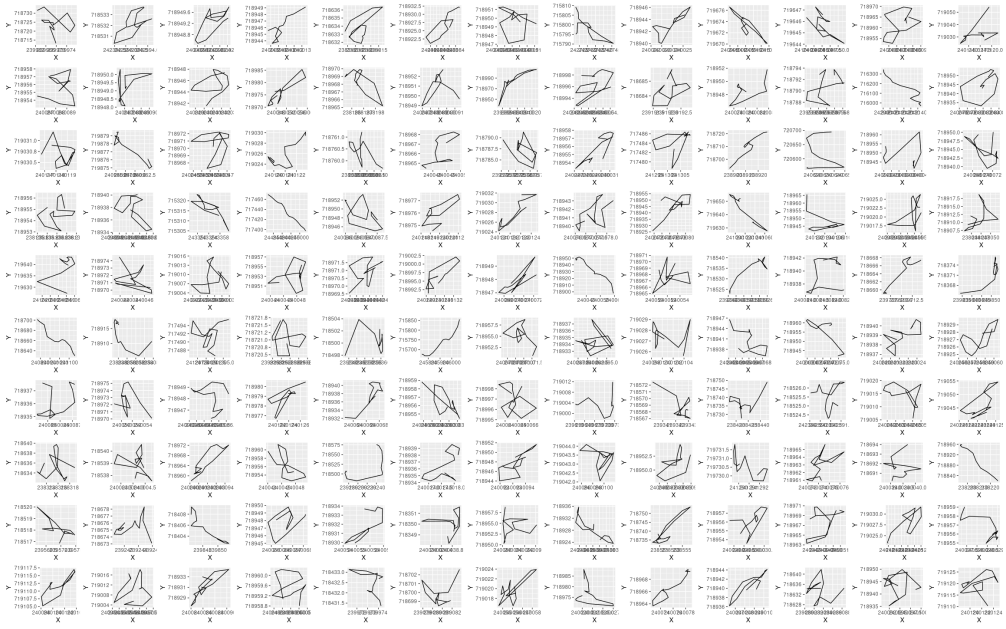

Figure A.23: 130 random juvenile barn owl segments from cluster C=6 (Table 2)

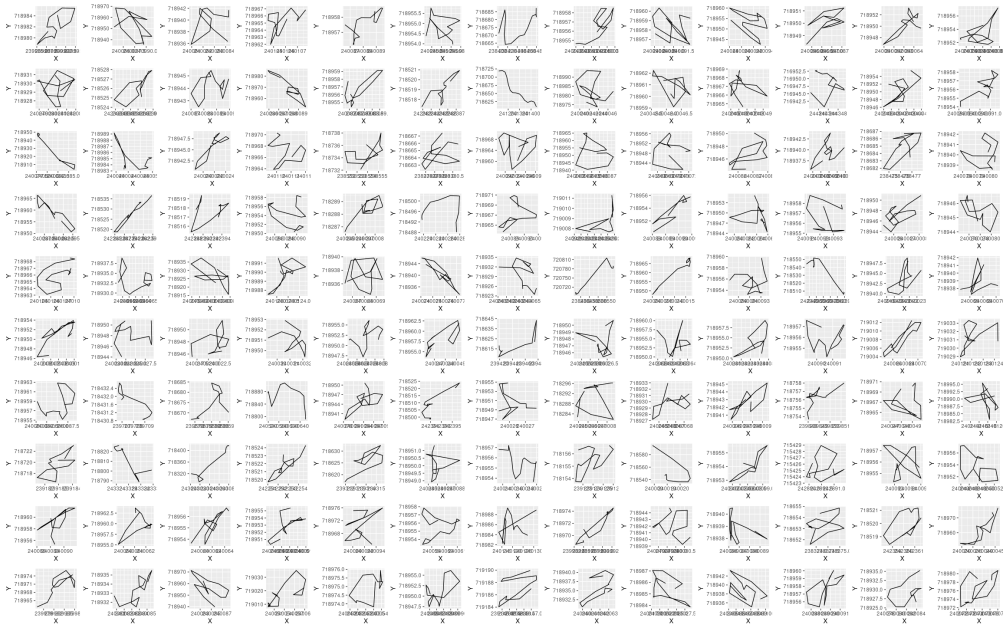

Figure A.24: 130 random juvenile barn owl segments from cluster C=7 (Table 2)

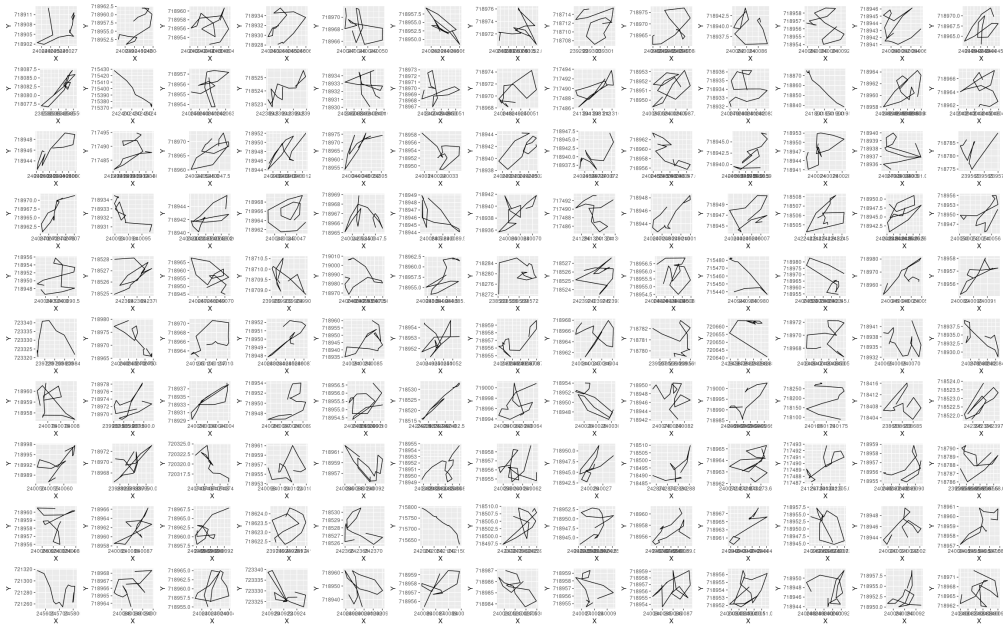

Figure A.25: 130 random juvenile barn owl segments from cluster C=8 (Table 2)

# B Numerical construction of a homogeneous movement map

A numerical representation of the mapping $\mathscr{F}_{\mu}^{\text{hom}}$ Eq 14, where we note that we have dropped the dependence on $\overline{\Delta\rho}$ because of the lack of circular bias in our movement simulations and the index $\mu$ in the image space) can be constructed as follows:

1. For a set of kernels defined in terms a positive integer $\hat{r}^{\max}$ and an incremental angle $\Delta\psi$ (typically equal to $\pi/12$ or $\pi/36$) define the following set of kernels

$$\mathscr{K}\left(\hat{r}^{\max},\Delta\psi\right)=\left\{K(r^{\min},r^{\max},\nu\Delta\psi)\,\middle|\,r^{\min}=0,\cdots,r^{\max},\ r^{\max}=1,\cdots,\hat{r}^{\max},\ \nu=1,\cdots,\frac{\pi}{\Delta\psi}\right\} \tag{B.6}$$

2. For each kernel in $\mathscr{K}\left(\hat{r}^{\max},\Delta\psi\right)$ run a simulation of the model, moving the individual over the desired homogeneous landscape for a selected period of period of time. For example, if one wants to generate 100 segments, each 15 points long, then the simulation will proceed for 1500 time steps.

3. For each of the simulated trajectories of the kernels in $\mathscr{K}\left(\hat{r}^{\max},\Delta\psi\right)$, carry out a segmentation with segment size $\mu$ and generate the selected set of segment statistics, in our case the values $\left(V,\text{SD}^{V},|\Delta\Theta|,\text{SD}^{|\Delta\Theta|}\right)$ for each segment. Compute the means of these statistics across the set of segments to obtain one candidate point of the mapping Eq 14.

4. In identifying the set of kernel $\left(r^{\min},r^{\max},\nu\Delta\psi\right)$ that generates an image under the mapping $\mathscr{F}_{\mu}^{\text{hom}}$ that is closest to the observed point $\left(\overline{V}^{\text{obs}},\overline{\text{SD}^{V}}^{\text{obs}},\overline{|\Delta\Theta|}^{\text{obs}},\overline{\text{SD}^{|\Delta\Theta|}}^{\text{obs}}\right)$ proceed as follows:

   (a) Discretize the image space of $\mathscr{F}_{\mu}^{\text{hom}}$ through a series of latin cube lattice computations of $\mathscr{F}_{\mu}^{\text{hom}}$ with arguments ranging over the cube with min and max bounding vertices $\left\{(0,0,0),(\hat{r^{\min}},\hat{r^{\max}},\hat{\nu}\Delta\psi)\right\}$ and intermediate cubes with min and max vertices $\left\{(r^{\min},r^{\max},\nu\Delta\psi),(r^{\min}+1,r^{\max}+1,(\nu+1)\Delta\psi)\right\}$, $r^{\min}=0,\cdots\hat{r^{\min}}-1$, $r^{\max}=0,\cdots\hat{r^{\max}}-1$, and $\nu=0,\cdots,\hat{\nu}-1$.

   (b) Select the intermediate cube that contains the image $\left(\overline{V}^{\text{obs}},\overline{\text{SD}^{V}}^{\text{obs}},\overline{|\Delta\Theta|}^{\text{obs}},\overline{\text{SD}^{|\Delta\Theta|}}^{\text{obs}}\right)$: i.e., the cube for which, making the $K_{\alpha}$ arguments in $\mathscr{F}_{\mu}^{\text{hom}}(K_{\alpha})$ explicity:

$$\mathscr{F}_{\mu}^{\text{hom}}\left(r^{\min},r^{\max},\nu\Delta\psi\right)\leq\left(\overline{V}^{\text{obs}},\overline{\text{SD}^{V}}^{\text{obs}},\overline{|\Delta\Theta|}^{\text{obs}},\overline{\text{SD}^{|\Delta\Theta|}}^{\text{obs}}\right)\leq\mathscr{F}_{\mu}^{\text{hom}}\left(r^{\min}+1,r^{\max}+1,(\nu+1)\Delta\psi\right) \tag{B.7}$$

   (c) Either select the vertex in the cube defined by $\left\{(r^{\min},r^{\max},\nu\Delta\psi),(r^{\min}+1,r^{\max}+1,(\nu+1)\Delta\psi)\right\}$ whose image under $\mathscr{F}_{\mu}^{\text{hom}}$ is closest to the observed point $\left(\overline{V}^{\text{obs}},\overline{\text{SD}^{V}}^{\text{obs}},\overline{|\Delta\Theta|}^{\text{obs}},\overline{\text{SD}^{|\Delta\Theta|}}^{\text{obs}}\right)$ or use a finer resolution latin cube grid within this cube to locate a set of parameters the provides an image as close as desired to the observed point.

# C ANIMOVER_1

## C.1 Output Data

At the end of the simulation the output data can either be automatically or manually saved (using a toggle switch on the console, see Fig 4, N) as a csv file. The header to this file contains a list of all the parameter and switch settings that were used to generate the run. In addition, the data are listed in 8 different columns depicted in Fig. C.1.

| 1 | Day | Delta | X | Y | Distance | Theta (Deg.) | Resources | Kernel |
|---|---|---|---|---|---|---|---|---|
| 656 | 1 | 654 | -76 | 104 | 10 | 0 | 133.74049 | BP: Movement |
| 657 | 1 | 655 | -86 | 104 | 10 | 180 | 133.88572 | WP: Rim |
| 658 | 1 | 656 | -96 | 104 | 10 | 180 | 133.88572 | BP: Movement |
| 659 | 1 | 657 | -87 | 107 | 9.486833 | 18.434949 | 134.171434 | WP: Rim |
| 660 | 1 | 658 | -87 | 107 | 0 | 0 | 134.427844 | BP: Vision |
| 661 | 1 | 659 | -87 | 107 | 0 | 0 | 134.641523 | WP: Consuming |
| 662 | 1 | 660 | -87 | 106 | 1 | 270 | 134.927237 | WP: Consuming |
| 663 | 1 | 661 | -87 | 105 | 1 | 270 | 135.180047 | WP: Consuming |
| 664 | 1 | 662 | -87 | 105 | 0 | 0 | 135.388468 | WP: Consuming |
| 665 | 1 | 663 | -86 | 105 | 1 | 0 | 135.548575 | WP: Consuming |
| 666 | 1 | 664 | -88 | 96 | 9.219544 | 257.471192 | 135.548575 | WP: Rim |
| 667 | 1 | 665 | -92 | 87 | 9.848858 | 246.037511 | 135.548575 | BP: Movement |
| 668 | 1 | 666 | -97 | 79 | 9.433981 | 237.994617 | 135.548575 | BP: Movement |
| 669 | 1 | 667 | -105 | 74 | 9.433981 | 212.005383 | 135.548575 | BP: Movement |
| 670 | 1 | 668 | -113 | 69 | 9.433981 | 212.005383 | 135.548575 | BP: Movement |
| 671 | 1 | 669 | -122 | 65 | 9.848858 | 203.962489 | 135.548575 | BP: Movement |
| 672 | 1 | 670 | -131 | 63 | 9.219544 | 192.528808 | 135.548575 | BP: Movement |
| 673 | 1 | 671 | -139 | 58 | 9.433981 | 212.005383 | 135.548575 | BP: Movement |
| 674 | 1 | 672 | -147 | 53 | 9.433981 | 212.005383 | 135.548575 | BP: Movement |
| 675 | 1 | 673 | 148 | 45 | 9.433981 | 237.994617 | 135.548575 | BP: Movement |
| 676 | 1 | 674 | 147 | 36 | 9.055385 | 263.659808 | 135.548575 | BP: Movement |
| 677 | 1 | 675 | 145 | 27 | 9.219544 | 257.471192 | 135.548575 | BP: Movement |
| 678 | 1 | 676 | 145 | 18 | 9 | 270 | 135.548575 | BP: Movement |
| 679 | 1 | 677 | 142 | 9 | 9.486833 | 251.565051 | 135.834289 | BP: Vision |
| 680 | 1 | 678 | 143 | 7 | 2.236068 | 296.565051 | 136.120003 | BP: Vision |
| 681 | 1 | 679 | 143 | 6 | 1 | 270 | 136.405718 | WP: Consuming |
| 682 | 1 | 680 | 142 | 6 | 1 | 180 | 136.691432 | WP: Consuming |
| 683 | 1 | 681 | 141 | 15 | 9.055385 | 96.340192 | 136.691432 | WP: Rim |
| 684 | 1 | 682 | 144 | 9 | 6.708204 | 296.565051 | 136.977146 | WP: Seeking |
| 685 | 1 | 683 | 144 | 7 | 2 | 270 | 137.26286 | BP: Vision |
| 686 | 1 | 684 | 143 | 7 | 1 | 180 | 137.519818 | WP: Consuming |
| 687 | 1 | 685 | 142 | 7 | 1 | 180 | 137.805532 | WP: Consuming |
| 688 | 1 | 686 | 142 | 8 | 1 | 90 | 138.091246 | WP: Consuming |
| 689 | 1 | 687 | 142 | 8 | 0 | 0 | 138.347656 | WP: Consuming |
| 690 | 1 | 688 | 142 | 9 | 1 | 90 | 138.605148 | WP: Consuming |
| 691 | 1 | 689 | 143 | 9 | 1 | 0 | 138.890863 | WP: Consuming |
| 692 | 1 | 690 | 143 | 10 | 1 | 90 | 139.176577 | WP: Consuming |
| 693 | 1 | 691 | 144 | 10 | 1 | 0 | 139.462291 | WP: Consuming |
| 694 | 1 | 692 | 144 | 10 | 0 | 0 | 139.718701 | WP: Consuming |

Figure C.1: The 8 columns in the csv data output file that can be saved at the end of the simulation as as follows: `Day`, `Delta` (within-day step), `X` ($x$-location), `Y` ($y$-location), `Distance` (distance moved), `Theta` (angle of heading in degrees), `Resources` (agent-state), `Kernel` (BP or WP). We note that one computes the turning angle $\Delta\theta$ at time $t$ by subtracting the angle of headings at time $t$ from the angle of heading at time $t-1$ taking into account that angles are specified modulo 360 degrees (e.g., if $\theta_{t-1} = 350$ and $\theta_t = 10$, then the turning angle is $\Delta\theta = 10 - (350 - 360) = 20$ degrees). Thus angles have heading range over $[-\pi, \pi]$. In addition, the third argument in our StaME is the average of the absolute values of the angles of heading rather than just the angles themselves. Also note that the BP movement mode (green entries) has two states: `Movement` (movement within the kernel rim which rim exists whenever $r_{\text{wp}}^{\text{min}} > 0$) or `Vision` (movement within the kernel sector, possibly between the inner rim radius and the current location). The WP movement mode has three states: `Consuming` (when moving to a resource-rich location), `Seeking` (an interim step in looking for a resource rich location when the kernel contains insufficient resources), or `Rim` moving the maximum step length ($r_{\text{bp}}^{\text{max}}$ rather than $r_{\text{wp}}^{\text{max}}$ when the seeking state fails to provide a suitable resource location). The indicted 10-point segmentation of the data shows how both pure and mixed movement segments arise during segmentation.

## C.2 Downloading and Running the App

ANIMOVER_1 and the most recent release of Numerus Studio can be downloaded without cost from the Numerus Webpage https://www.numerusinc.com/studio/. Installers for Numerus Studio are provided for Mac and Windows platforms. Instructions for using Numerus Studio are contained in the RAMP Users Guide at https://wiki.numerusinc.com/index.php/Ramp_User_Guide

ANIMOVER_1 is deployed in the RAMP file `Ani1Cr3.nms`. After installing Numerus studio, open this file and launch it from the Studio launchpad. Documentation for this RAMP can be found at https://wiki.numerusinc.com/index.php/Animover_1.