

momentuHMM: R package for analysis of telemetry data using generalized multivariate hidden Markov models of animal movement

Brett T. McClintock¹ and Théo Michelot²

¹Marine Mammal Laboratory
Alaska Fisheries Science Center
NOAA National Marine Fisheries Service
Seattle, U.S.A.
Email: brett.mcclintock@noaa.gov

²School of Mathematics and Statistics
University of Sheffield
Sheffield, U.K.

RUNNING HEAD: R package **momentuHMM**

May 1, 2017

Summary

1. Discrete-time hidden Markov models (HMMs) have become an immensely popular tool for inferring latent animal behaviors from telemetry data, largely because they are relatively fast and easy to implement when data streams are observed without error and at regular time intervals. While movement HMMs typically rely solely on location data, auxiliary biotelemetry and environmental data are powerful and readily-available resources for incorporating much more behavioral realism and inferring ecological relationships that would otherwise be difficult or impossible to infer from location data alone. However, there is a paucity of generalized user-friendly software available for implementing (multivariate) HMMs of animal movement. Furthermore, location measurement error, temporal irregularity, and other forms of missing data are often pervasive in telemetry studies (particularly in marine systems).

2. Here we introduce an open-source R package, `momentuHMM`, that addresses many of the deficiencies in existing software. Features for multivariate HMMs in `momentuHMM` include: 1) tools for data pre-processing and visualization; 2) user-specified probability distributions for an unlimited number of data streams, such as those based on location (e.g., step length, turning angle) and auxiliary biotelemetry data (e.g., from pressure, conductivity, heart rate, or motion sensors); 3) user-specified design matrices and constraints for covariate modelling of state transition probability and probability distribution parameters using linear model formulas familiar to most R users; 4) multiple imputation methods that account for observation error attributable to measurement error and temporally-irregular or missing data; 5) seamless integration of spatio-temporal environmental covariate data (e.g., wind direction, forest cover,

24 sea ice concentration) using rasters; 6) incorporation of “activity center” effects on
25 parameters (e.g., areas associated with attractive or repulsive forces); 7) circular-
26 circular regression models for angular probability distributions; 8) cosinor and spline
27 regression formulas for cyclical (e.g., daily, seasonal) and other complicated patterns;
28 and 9) data simulation capabilities for power analyses and assessing model perfor-
29 mance, including simulation of location data subject to temporal irregularity and/or
30 measurement error.

31 **3.** After providing a brief introduction to (multivariate) HMMs for telemetry data,
32 we demonstrate some of the capabilities of `momentuHMM` using real-world examples.
33 This brief tutorial includes workflows for data formatting, model specification, model
34 fitting, and diagnostics.

35 **4.** While many of the features of `momentuHMM` were motivated by animal movement
36 data, the package can be used for analyzing any type of data that is amenable to
37 (multivariate) HMMs. Practitioners interested in additional features for `momentuHMM`
38 are encouraged to contact the authors.

39

40 **Key-words** biologging, biotelemetry, `crawl`, `moveHMM`, state-space model, state-
41 switching

42 **1 Introduction**

43 Discrete-time hidden Markov models (HMMs) have become immensely popular for
44 the analysis of animal telemetry data (e.g. Morales *et al.* 2004; Jonsen *et al.* 2005;
45 Langrock *et al.* 2012; McClintock *et al.* 2012). In short, an HMM is a time series
46 model composed of one or more observable data streams $(\mathbf{Z}_1, \dots, \mathbf{Z}_T)$, each of which

is generated by N state-dependent probability distributions, where the unobservable (hidden) state sequence $(S_t \in \{1, \dots, N\}, t = 1, \dots, T)$ is assumed to be a Markov chain. The state sequence of the Markov chain is governed by (typically first-order) state transition probabilities, $\gamma_{ij}^{(t)} = \Pr(S_{t+1} = j \mid S_t = i)$ for $i, j = 1, \dots, N$, and an initial distribution $\boldsymbol{\delta}^{(0)}$. The likelihood of an HMM can be succinctly expressed using the forward algorithm:

$$\mathcal{L} = \boldsymbol{\delta}^{(0)} \boldsymbol{\Gamma}^{(1)} \mathbf{P}(\mathbf{z}_1) \boldsymbol{\Gamma}^{(2)} \mathbf{P}(\mathbf{z}_2) \boldsymbol{\Gamma}^{(3)} \dots \boldsymbol{\Gamma}^{(T-1)} \mathbf{P}(\mathbf{z}_{T-1}) \boldsymbol{\Gamma}^{(T)} \mathbf{P}(\mathbf{z}_T) \mathbf{1}^N \quad (1)$$

where $\boldsymbol{\Gamma}^{(t)} = \left(\gamma_{ij}^{(t)} \right)$ is the $N \times N$ transition probability matrix, $\mathbf{P}(\mathbf{z}_t) = \text{diag}(p_1(\mathbf{z}_t), \dots, p_N(\mathbf{z}_t))$, $p_s(\mathbf{z}_t)$ is the conditional probability density of \mathbf{Z}_t given $S_t = s$, and $\mathbf{1}^N$ is a N -vector of ones (for a thorough introduction to HMMs see Zucchini *et al.* 2016).

One of the most common discrete-time animal movement HMMs for telemetry location data is composed of two data streams, step length and turning angle (or bearing), which are calculated for each of the T time steps from the temporally-regular observations of an animal's position, (x_t, y_t) , for $t = 1, \dots, T+1$ (e.g. Morales *et al.* 2004; Langrock *et al.* 2012; McClintock *et al.* 2012). Step length (l_t) is typically calculated as the Euclidean distance between the locations (x_t, y_t) and (x_{t+1}, y_{t+1}) , while turning angle (ϕ_t) is calculated as the change in bearing ($b_t = \text{atan2}(y_{t+1} - y_t, x_{t+1} - x_t)$) between the intervals $[t-1, t]$ and $[t, t+1]$ (e.g. $\phi_t = 0$ if $b_{t-1} = b_t$). For this HMM composed of 2 data streams, $\mathbf{z}_t = (l_t, \phi_t)$, and, conditional on the latent state S_t , independent probability distributions are typically assumed for each stream, e.g., $p_s(\mathbf{z}_t) = p_s(l_t)p_s(\phi_t)$. Some common probability distributions for the step length data stream are the Gamma or Weibull distributions, while the wrapped Cauchy or von Mises distributions are often employed for turning angle or bearing.

While HMMs for animal movement based solely on location data are somewhat limited in the number and type of biologically-meaningful movement behavior states they are able to accurately identify, advances in biologging technology are now allowing the collection of valuable auxiliary biotelemetry data (e.g., dive activity, accelerometer, heart rate, stomach temperature), which, when combined with location data, allow for multivariate HMMs that can incorporate much more behavioral realism and facilitate inferences about complex ecological relationships that would otherwise be difficult or impossible to infer from location data alone (e.g. McClintock *et al.* 2013; DeRuiter *et al.* 2017; McClintock *et al.* 2017). Multivariate HMMs that utilize both location and auxiliary biotelemetry data can facilitate the identification of additional states that go beyond the $N = 2$ state approaches that are most frequently used by practitioners. For example, the most widely used 2-state HMMs for animal movement include “encamped” (or “foraging”) and “exploratory” (or “transit”) states characterized by area-restricted-search-type movements (shorter step lengths with little to no directional persistence) and migratory-type movements (longer step lengths with high directional persistence), respectively (Morales *et al.* 2004; Jonsen *et al.* 2005). However, very different behaviors can exhibit similar horizontal trajectories. For example, for herbivores such as North American elk (Morales *et al.* 2004) or central-place foragers such as harbour seals (McClintock *et al.* 2013), the horizontal trajectories of “resting” and “foraging” movements can be very difficult to distinguish. Standard 2-state HMMs based solely on horizontal trajectory will tend to lump these behaviors together, and this could have unintended consequences if, for example, one intends to use the estimated state sequences to identify foraging habitat. In order to tweeze apart distinct behaviors with similar horizontal

trajectories, additional states informed by auxiliary information (such as mandible accelerometer or dive data) must often be incorporated as additional data stream(s) in a multivariate HMM.

When data streams are observed without error and at regular time intervals, a major advantage of HMMs is the relatively fast and efficient maximization of the likelihood using the forward algorithm (Eq. 1). However, location measurement error is rarely non-existent in animal-borne telemetry studies and depends on both the device and the system under study. For example, GPS errors are typically less than 50m, but Argos errors can exceed 10km (e.g. Costa *et al.* 2010). An extreme case of missing data can arise when location data are obtained with little or no temporal regularity, as in many marine mammal telemetry studies (e.g. Jonsen *et al.* 2005), such that few (if any) observations align with the regular time steps required by discrete-time HMMs. When explicitly accounting for uncertainty attributable to location measurement error, temporally-irregular observations, or other forms of missing data, one must typically fit (multivariate) HMMs using computationally-intensive (and often time-consuming) model fitting techniques such as Markov chain Monte Carlo (Jonsen *et al.* 2005; McClintock *et al.* 2012). However, complex analyses requiring novel statistical methods and custom model-fitting algorithms are not practical for many practitioners.

While statisticians have been applying HMMs to telemetry data for decades, R (R Core Team 2016) packages such as **bsam** (Jonsen *et al.* 2005), **moveHMM** (Michelet *et al.* 2016), and **swim** (Whoriskey *et al.* 2017) have recently helped make these models of animal movement behavior more accessible to the practitioners that are actually conducting telemetry studies. These advances represent important steps

117 toward making HMMs of animal movement more accessible, but the models that
 118 can currently be implemented using existing software remain limited in many key
 119 respects. For example, existing HMM software for animal movement is limited to
 120 two data streams based solely on location data (e.g. step length and turning an-
 121 gle), and while `moveHMM` allows for a user-specified number of latent behavioral states
 122 (`bsam` and `swim` are limited to $N = 2$ states), it is typically difficult to identify
 123 >2 biologically-meaningful behavior states from only 2 data streams (e.g. Morales
 124 *et al.* 2004; Beyer *et al.* 2013; McClintock *et al.* 2014). Both `moveHMM` and `swim`
 125 are designed for temporally-regular (or linearly-interpolated) location data with neg-
 126 ligible measurement error, but the realities of animal-borne telemetry often yield
 127 temporally-irregular location data subject to error (particularly in aquatic environ-
 128 ments). Other notable deficiencies of existing software include limited abilities to
 129 incorporate spatio-temporal environmental or individual covariates on parameters,
 130 biased (or directed) movements in response to attractive or repulsive forces (e.g. Mc-
 131 Clintock *et al.* 2012; Langrock *et al.* 2014), cyclical (e.g. daily, seasonal) and other
 132 more complicated behavioral patterns, or constraints on parameters.

133 To address these deficiencies in existing software, we introduce a new user-friendly
 134 R package, `momentuHMM` (Maximum likelihood analysis Of animal MovemENT behav-
 135 ior Using multivariate Hidden Markov Models), intended for practitioners wishing
 136 to implement more flexible and realistic (multivariate) HMM analyses of animal
 137 movement while accounting for common challenges associated with telemetry data.
 138 Features for multivariate HMM analyses in `momentuHMM` include: 1) tools for data
 139 pre-processing and visualization; 2) user-specified probability distributions for an un-
 140 limited number of data streams and latent behavior states; 3) user-specified design

matrices and constraints for covariate modelling of state transition probability and probability distribution parameters using linear model formulas familiar to most R users; 4) multiple imputation methods that account for observation error attributable to measurement error and temporally-irregular or missing data (Hooten *et al.* 2017; McClintock 2017); 5) seamless integration of spatio-temporal environmental covariate data (e.g., wind direction, forest cover, sea ice concentration) using the **raster** package (Hijmans 2016); 6) incorporation of “activity center” effects (e.g., areas associated with attractive or repulsive forces); 7) circular-circular regression models for angular probability distributions (Duchesne *et al.* 2015); 8) cosinor (e.g. Cornelissen 2014) and spline regression formulas for cyclical and other complicated behavioral patterns; and 9) data simulation capabilities for power analyses and assessing model performance, including simulation of location data subject to temporal irregularity and/or measurement error.

In the following tutorial, we demonstrate some of the capabilities of **momentuHMM** using real-world examples, including an example of periodic cycles in African elephant movement, a 3-state (“resting”, “foraging”, “transit”) northern fur seal example incorporating auxiliary dive activity data (McClintock *et al.* 2014), a loggerhead turtle example relating “foraging” and “transit” movements to ocean surface currents, and a 5-state grey seal example incorporating biased movements toward haul-out and foraging locations (McClintock *et al.* 2012). This brief tutorial includes workflows for data formatting, model specification, model fitting, and diagnostics. While many of the features of **momentuHMM** were motivated by animal movement data, the package can be used for analyzing any type of data that is amenable to (multivariate) HMMs. Additional information, including help files, data, examples, and package us-

age is available by downloading the `momentuHMM` package from CRAN (<http://cran.r-project.org>) or github (<https://github.com/bmcclintock/momentuHMM>). This article describes `momentuHMM` version 1.0.0.

2 `momentuHMM` overview

Before delving into some of the finer details, we will first provide an overview of the main features and functions of the package. While space is limited in this tutorial, further details on implementation can be found in the package’s documentation and vignette. The workhorse functions of `momentuHMM` are listed in Table 1. Usage of several of these functions (e.g. `fithHMM`, `prepData`, `simData`) is deliberately very similar to equivalent functions in `moveHMM` (Michelot *et al.* 2016), but the `momentuHMM` arguments for these functions have been generalized and expanded to accommodate a more flexible framework for data pre-processing, model specification, parameterization, and simulation. R users already familiar with `moveHMM` will therefore likely find it easy to immediately begin using `momentuHMM`.

One of the key features of `momentuHMM` is the ability to specify an unlimited number of HMM data streams from a broad range of commonly used probability distributions. Any of the probability distribution parameters (as well as the state transition probabilities) can be modelled as a function of environmental and individual covariates using link functions (Table 2). For any given “natural scale” (or “real scale”) probability distribution parameter θ , all of the link functions (g) in `momentuHMM` are of the general form $g(\theta) = \mathbf{X}_\theta \boldsymbol{\beta}_\theta$, where \mathbf{X}_θ is the $T \times k$ design matrix (composed of k covariates) and $\boldsymbol{\beta}_\theta$ is the corresponding k -vector of “working scale” (or “beta scale”) parameters for θ . For example, suppose step length is as-

Table 1. Workhorse functions for the R package `momentuHMM`.

Function	Description
<code>crawlMerge</code>	Merge <code>crawlWrap</code> output with additional data streams or covariates
<code>crawlWrap</code>	Fit <code>crawl</code> models and predict temporally-regular locations
<code>fitHMM</code>	Fit a (multivariate) HMM to the data
<code>MIfitHMM</code>	Fit (multivariate) HMMs to multiple imputation data
<code>MIpool</code>	Pool <code>momentuHMM</code> model results across multiple imputations
<code>plot.crwData</code>	Plot <code>crawlWrap</code> output
<code>plot.miSum</code>	Plot summaries of multiple imputation <code>momentuHMM</code> models
<code>plot.momentuHMM</code>	Plot summaries of <code>momentuHMM</code> models
<code>plot.momentuHMMData</code>	Plot summaries of selected data streams and covariates
<code>plotPR</code>	Plot time series, qq-plots and sample ACFs of pseudo-residuals
<code>plotSat</code>	Plot locations on satellite image
<code>plotSpatialCov</code>	Plot locations on raster image
<code>plotStates</code>	Plot the (Viterbi-)decoded states and state probabilities
<code>prepData</code>	Pre-process data streams and covariates
<code>pseudoRes</code>	Calculate pseudo-residuals for <code>momentuHMM</code> models
<code>simData</code>	Simulate movement data using a (multivariate) HMM
<code>stateProbs</code>	State probabilities for each time step
<code>viterbi</code>	Most likely state sequence (using the Viterbi algorithm)

188 sumed to have a Gamma distribution, $l_t \mid S_t = s \sim \text{Gamma}(\mu_s, \sigma_s)$. In `momentuHMM`,
 189 the natural scale parameters for the Gamma distribution are the (state-dependent)
 190 step length mean ($\mu_s > 0$) and standard deviation ($\sigma_s > 0$). Because both of these
 191 parameters must be positive, the log link function is a natural choice for modelling
 192 these parameters as a function of covariates, e.g., $\log(\boldsymbol{\mu}) = \mathbf{X}_\mu \boldsymbol{\beta}_\mu$.

193 2.1 Data preparation and visualization

194 For temporally-regular location data with negligible measurement error, the `prepData`
 195 function is used to create a `momentuHMMData` object that can be used for data visu-
 196 alization and further analysis. The arguments for `prepData` include:

- 197 • **Data** A data frame with $T + 1$ rows including optionally a field ‘ID’ (identi-
 198 fiers for different individuals), coordinates from which step length (‘step’) and
 199 turning angle (‘angle’) data streams are to be calculated, any additional data
 200 streams, and any covariates identified in the `covNames` and `angleCovs` argu-
 201 ments. Alternatively, **Data** can be a `crwData` object returned by `crawlWrap`.
- 202 • **type** Coordinate type; ‘UTM’ if easting-northing or ‘LL’ if longitude-latitude.
- 203 • **coordNames** Names of the two coordinate columns in **Data**. If `coordNames=NULL`
 204 then step lengths, turning angles, and any location-based covariates (i.e., those
 205 specified by `spatialCovs`, `centers`, and `angleCovs`) are not calculated.
- 206 • **covNames** Character vector indicating the names of any covariates in **Data**.
 207 Any variables in **Data** (other than “ID”) that are not identified in `covNames` or
 208 `angleCovs` are assumed to be data streams.

Table 2. Data stream (z) probability distributions, natural parameters, and default link functions for covariate modelling. Probability distributions with positive support can be zero-inflated (with additional zero-mass parameters), while the Beta distribution can be zero- and one-inflated (with additional one-mass parameters). If user-specified bounds are provided, then custom link functions are used instead of the defaults (see package documentation for further details). If circular-circular regression is specified for the mean of angular distributions (“vm” and “wrpcauchy”), then the link function described in Duchesne *et al.* (2015) is used. Users seeking additional probability distributions are encouraged to contact the authors.

Distribution	Support	Parameters	Link function
Beta (“beta”)	$z_t \in (0, 1)$	shape1 > 0 shape2 > 0 zero-mass $\in (0, 1)$ one-mass $\in (0, 1)$	log log logit logit
Exponential (“exp”)	$z_t > 0$	rate > 0 zero-mass $\in (0, 1)$	log logit
Gamma (“gamma”)	$z_t > 0$	mean > 0 sd > 0 zero-mass $\in (0, 1)$	log log logit
Log normal (“lnorm”)	$z_t > 0$	location $\in \mathbb{R}$ scale > 0 zero-mass $\in (0, 1)$	identity log logit
Poisson (“pois”)	$z_t \in \{0, 1, \dots\}$	lambda > 0	log
Von Mises (“vm”)	$z_t \in (-\pi, \pi]$	mean $\in (-\pi, \pi]$ concentration > 0	tan(mean/2) log
Weibull (“weibull”)	$z_t > 0$	shape > 0 scale > 0	log log
Wrapped Cauchy (“wrpcauchy”)	$z_t \in (-\pi, \pi]$	mean $\in (-\pi, \pi]$ concentration $\in (0, 1)$	tan(mean/2) logit

- 209 • **spatialCovs** List of Raster-class objects (Hijmans 2016) containing spatio-
 210 temporally referenced covariates. Covariates specified by **spatialCovs** are ex-
 211 tracted from the raster layer(s) based on the location data. Raster stacks may
 212 also be included, in which case the appropriate z values (e.g. time, date) must
 213 also be included in **Data**.
- 214 • **centers** 2-column matrix providing the coordinates for any activity centers
 215 (e.g., potential centers of attraction or repulsion) from which distance and
 216 angle covariates will be calculated based on the location data and returned in
 217 the **momentuHMMData** object.
- 218 • **angleCovs** Character vector indicating the names of any circular-circular re-
 219 gression angular covariates in **Data** or **spatialCovs** that need conversion from
 220 standard direction (in radians relative to the x-axis) to turning angle (relative
 221 to previous movement direction).

222 Summary plots of the **momentuHMMData** object returned by **prepData** can be created
 223 for any data stream or covariate using the generic **plot** function.

224 If location data are temporally-irregular or subject to measurement error, then
 225 they are not suitable for **prepData**. In this case, **momentuHMM** can be used to perform a
 226 2-stage multiple imputation approach (McClintock 2017). We discuss this pragmatic
 227 approach to incorporating uncertainty attributable to observation error and temporal
 228 irregularity into multivariate HMM analyses in section 2.3.

2.2 HMM specification and fitting

Once a `momentuHMMData` object has been created using `prepData`, then the data are ready to be passed to the generalized multivariate HMM-fitting function `fitHMM`. There are many different options for specifying HMMs using `fitHMM`, so here we will only focus on several of the most important and useful features (further details of all `fitHMM` arguments are in the package documentation). The bare essentials of `fitHMM` include the arguments:

- `data` A `momentuHMMData` object
- `nbStates` Number of latent states (N)
- `dist` A named list indicating the probability distributions of the data streams.
- `estAngleMean` An optional named list indicating whether or not to estimate the angle mean for data streams with angular distributions (e.g. turning angle). If not estimated (the default), the angle mean is fixed to 0.
- `formula` Regression formula for the transition probability covariates
- `stationary` Logical indicating whether or not the initial distribution is considered equal to the stationary distribution (must be `FALSE` if `formula` includes covariates)
- `Par0` A named list containing vectors of starting values for the state-dependent probability distribution parameters of each data stream

These seven arguments are all that are needed in order to fit the HMMs currently supported in `moveHMM` (Michelot *et al.* 2016). For example, here is how the analysis

250 of 15 “wild haggis” tracks described in Michelot *et al.* (2016) would be implemented

251 using `momentuHMM`:

```
> library(momentuHMM)
> ### Load raw data
> rawHaggis<-read.csv("rawHaggises.csv")
> ### Process data
> processedHaggis<-prepData(data=rawHaggis,covNames=c("slope","temp"))
> ### Fit HMM
> # initial step distribution natural scale parameters
> stepPar0 <- c(1,5,0.5,3) # (mu_1,mu_2,sd_1,sd_2)
> # initial angle distribution natural scale parameters
> anglePar0 <- c(0,0,1,8) # (mean_1,mean_2,concentration_1,concentration_2)
> fitHaggis <- fitHMM(data = processedHaggis, nbStates = 2,
+                     dist = list(step = "gamma", angle = "vm"),
+                     Par0 = list(step = stepPar0, angle = anglePar0),
+                     formula = ~ slope + I(slope^2),
+                     estAngleMean = list(angle=TRUE))
```

252 Note that many of the arguments in `fitHMM` are lists, with each element of the
253 list corresponding to a data stream. The list names provided in `dist`, `Par0`, and
254 `estAngleMean` (e.g. ‘step’ and ‘angle’) must therefore have a corresponding column
255 in `data` with the same name. Additional data streams can be added to the model
256 by simply adding the additional elements to these list arguments (see example in
257 section 3.2).

258 As seen above, the `formula` argument can include many of the functions and
259 operators commonly used to construct terms in R linear model formulas (e.g. `a*b`,
260 `a:b`, `cos(a)`). Unique to `momentuHMM`, the `formula` argument can also be used to
261 specify transition probability matrix models that incorporate cyclical patterns (using
262 the `cosinor` special function; see example in section 3.1), splines for explaining
263 other more complicated patterns (e.g., `bs` and `ns` functions in the R base package
264 `splines`), and factor variables (e.g., `formula=~ID` for individual-level effects). By

265 default the `formula` argument applies to all state transition probabilities, but the
 266 special functions `state` and `betaCol` allow for state- and parameter-specific formulas
 267 to be specified. Specific state transition probabilities can also be fixed to zero (or
 268 any other value) using the `fixPar` argument, which can be useful for incorporating
 269 more behavioral realism. For example, `fixPar` can be used to prohibit or enforce
 270 switching from one particular state to another (possibly as a function of spatio-
 271 temporal covariates).

272 Similar to the `formula` argument for state transition probability modelling, it is
 273 through the `DM` argument of `fitHMM` that models are specified for the state-dependent
 274 probability distribution parameters for each data stream. `DM` is a list argument con-
 275 taining an element for each data stream, but each element itself is also a list specifying
 276 the design matrix formulas for each parameter. For example, the following fits the
 277 exact same wild haggis model as above, but employs a user-specified (intercept-only)
 278 design matrix for the step length data stream:

```
> stepDM <- list(mean = ~1, sd = ~1)
> ### Fit HMM          using user-specified DM
> fitHaggisDM <- fitHMM(data = processedHaggis, nbStates = 2,
+                       dist = list(step = "gamma", angle = "vm"),
+                       DM = list(step = stepDM),
+                       Par0 = list(step = log(stepPar0), angle = anglePar0),
+                       formula = ~ slope + I(slope^2),
+                       estAngleMean = list(angle=TRUE))
```

279 Note that when `DM` is specified for a data stream, the initial parameter values (`Par0`)
 280 for that data stream now correspond to columns of the resulting design matrix and
 281 must be on the working scale instead of the natural scale. In this case, because the
 282 log link is used for the natural parameters of the Gamma distribution, `Par0$step`
 283 was specified on the log scale. The functions `getPar0` and `getParDM` are designed to

284 assist users in the specification of design matrices and corresponding initial values
 285 on the working scale for any given model (see package documentation for further
 286 details). DM formulas are just as flexible as the `formula` argument and, in addition
 287 to common linear model formula functions and operators, can also include cyclical
 288 cosinor models, splines, factor variables, and state-specific probability distribution
 289 parameter formulas (see examples in sections 3.3 and 3.4). As with the state transi-
 290 tion probabilities, working parameters for probability distributions can also be fixed
 291 to user-specified values using the `fixPar` argument.

292 Specification of design matrices using DM is not limited to formulas. Alternatively,
 293 “pseudo-design” matrices can be specified, using an R matrix with rows corresponding
 294 to the natural parameters and columns corresponding to the working parameters.
 295 The elements in the matrix may be numeric or character strings containing model
 296 formula terms (see examples in sections 3.2 and 3.4). Using a pseudo-design matrix
 297 for step length, the following is yet another way to implement the exact same wild
 298 haggis model:

```
> stepDMp <- matrix(c(1,0,0,0,
+                     0,1,0,0,
+                     0,0,1,0,
+                     0,0,0,1),4,4,byrow=TRUE)
> rownames(stepDMp) <- c("mean_1","mean_2","sd_1","sd_2")
> colnames(stepDMp) <- c("mean_1:(Intercept)","mean_2:(Intercept)",
+                        "sd_1:(Intercept)","sd_2:(Intercept)")
> ### Fit HMM using user-specified DM
> fitHaggisDMp <- fitHMM(data = processedHaggis, nbStates = 2,
+                        dist = list(step = "gamma", angle = "vm"),
+                        DM = list(step = stepDMp),
+                        Par0 = list(step = log(stepPar0), angle = anglePar0),
+                        formula = ~ slope + I(slope^2),
+                        estAngleMean = list(angle=TRUE))
```

299 (note that column and rownames for pseudo-design matrices are not required but
 300 can be useful). Pseudo-design matrices allow for the sharing of common working
 301 parameters (such as intercept terms) among natural scale parameters, and this can be
 302 used to constrain natural scale parameters (e.g., $\mu_1 \leq \mu_2$) when used in tandem with
 303 the `cons` argument. This is particularly useful for preventing state label switching
 304 when repeatedly fitting the same HMM using multiple imputation methods (see
 305 section 2.3).

306 Another noteworthy `fitHMM` argument, `circularAngleMean`, is a list argument
 307 that enables users to specify circular-circular regression models for the mean (μ)
 308 parameter of angular distributions, such as the wrapped Cauchy and von Mises,
 309 instead of circular-linear models based on the tangent link function (Table 2). When
 310 `circularAngleMean` is specified as `TRUE` for any given angular data stream (e.g.
 311 turning angle), then a special link function based on Duchesne *et al.* (2015) is used:

$$\boldsymbol{\mu} = \text{atan2}(\sin(\mathbf{X}_\mu)\boldsymbol{\beta}_\mu, 1 + \cos(\mathbf{X}_\mu)\boldsymbol{\beta}_\mu), \quad (2)$$

312 where \mathbf{X}_μ is a $T \times k$ matrix composed of the turning angles between k angular
 313 covariates (e.g., wind direction, sea surface currents) and the bearing of movement
 314 during the previous time step, i.e., each element $x_{t,i} = \text{atan2}(\sin(r_{t,i} - b_{t-1}), \cos(r_{t,i} -$
 315 $b_{t-1}))$ for angular covariate $r_{t,i}$ and $i = 1, \dots, k$ (note that `prepData` calculates
 316 \mathbf{X}_μ based on the `angleCovs` argument so users need not bother). Because this
 317 link function is designed for turning angles, a turning angle of 0 is provided as the
 318 reference angle (hence the “1+” preceeding the cosine term in Eq. 2). Thus as a
 319 trade-off between biased and correlated movements, the working parameters ($\boldsymbol{\beta}_\mu$) for
 320 the expected turning angle at time t weight the attractive (or repulsive) strengths of

the angular covariates relative to directional persistence. When all $\beta_\mu = 0$, the model reduces to a correlated random walk, but an increasingly biased random walk results as β_μ gets larger (or smaller). Many interesting hypotheses about animal movement can be addressed using circular-circular regression on movement direction, including the effects of wind, sea surface currents (see example in section 3.3), and centers of attraction or repulsion (see example in section 3.4)

2.3 Multiple imputation

When location data are temporally-irregular or subject to measurement error, then they are not suitable for standard maximum-likelihood HMM analyses based on the forward algorithm (Eq. 1). In this case, `momentuHMM` can be used to perform the 2-stage multiple imputation approach of McClintock (2017). The basic concept is to first employ a single-state (i.e., $N = 1$) movement model that is relatively easy to fit but can accommodate location measurement error and temporally-irregular or missing observations (e.g. Johnson *et al.* 2008). The second stage involves repeatedly fitting the desired HMM to m temporally-regular realizations of the position process drawn from the model output of the first stage. Data streams or covariates that are dependent on location (e.g., step length, turning angle, habitat type, snow depth, sea surface temperature) will of course vary among the m realizations of the position process, and the pooled inferences across the HMM analyses therefore reflect location uncertainty.

There are three primary functions (`MifitHMM`, `MIpool`, and `crawlWrap`) for performing multiple imputation HMM analyses in `momentuHMM`, and all rely on parallel processing to speed up computations. `crawlWrap` is a wrapper function for fitting the

344 continuous-time correlated random walk (CTCRW) model of Johnson *et al.* (2008)
 345 to one or more tracks (subject to location measurement error and/or temporal ir-
 346 regularity) and then predicting temporally-regular tracks of the user's choosing (e.g.
 347 15 min, hourly, daily) based on the CTCRW model output. `crawlWrap` returns a
 348 `crwData` object that can be used to draw m realization of the position process within
 349 the `MifitHMM` function. `MifitHMM` is essentially a wrapper function for `fitHMM` that
 350 repeatedly fits the same user-specified HMM to m imputed data sets and stores the
 351 output from each of the m model fits. If a `crwData` object is provided, then `MifitHMM`
 352 will first draw m imputations based on the `crwData` output and then fit the speci-
 353 fied HMM to each imputed data set. If users wish to use a movement model other
 354 than the CTCRW to account for measurement error and temporal irregularity (e.g.
 355 Calabrese *et al.* 2016; Gurarie *et al.* 2017), or if other observation error processes
 356 (e.g. missing data) are to be accounted for in the imputation step, `MifitHMM` can
 357 also be used for analysis of a list of m `momentuHMMData` objects that were imputed
 358 by the user. Based on the m model fits, the `Mipool` function calculates pooled es-
 359 timates, standard errors, and confidence intervals for the working scale parameters,
 360 natural scale parameters (based on transformations of the pooled working param-
 361 eters and mean or user-specified values for any covariates), state sequences, state
 362 probabilities, and activity budgets (i.e. the proportion of the T times step assigned
 363 to each state) using standard multiple imputation formulae (Rubin & Schenker 1986;
 364 McClintock 2017). `Mipool` can be called separately or within `MifitHMM` (using the
 365 `poolEstimates` argument), and the function returns a `miSum` object containing the
 366 pooled output across all imputatons. See sections 3.2, 3.3, and 3.4 for example HMM
 367 analyses that use multiple imputation to account for location measurement error and

368 temporally irregularity.

369 2.4 Model visualization and diagnostics

370 The generic `plot` functions for `momentuHMM` models (`plot.momentuHMM` and `plot.miSum`)
371 plot the data stream histograms along with their corresponding estimated probability
372 distributions, the estimated natural parameters and state transition probabilities as
373 a function of any covariates included in the model, and the tracks of all individuals
374 (color-coded by the most likely state sequence). By default, the probability distri-
375 butions are plotted based on the means of any covariate values, but user-specified
376 covariate values for the plots can be provided using the `covs` argument. When the ar-
377 gument `plotCI=TRUE`, then confidence intervals for the natural parameters and state
378 transition probabilities are also plotted. Confidence intervals are calculated from the
379 working parameter estimates based on the delta method and finite-difference approxi-
380 mations of the first derivative for the transformation using the `numDeriv::grad` func-
381 tion (Gilbert & Varadhan 2016). For multiple imputation analyses (`plot.miSum`), all
382 plots are based on the pooled parameter estimates and the means of any covariates
383 (if not provided by the `covs` argument) across each imputation. Using the argument
384 `errorEllipse`, `plot.miSum` will include estimated location error ellipses in the plots
385 of individual tracks. The functions `plotSat`, `plotSpatialCov`, and `plotStates` (Ta-
386 ble 1) provide further methods for visualizing model results.

387 Diagnostic tools include the calculation and plotting of pseudo-residuals (Zucchini
388 *et al.* 2016) using the `pseudoRes` and `plotPR` functions, respectively. Akaike’s Infor-
389 mation Criterion can be calculated for one or more models using the `AIC.momentuHMM`
390 function.

391 2.5 Simulation

392 The function `simData` can be used to simulate multivariate HMM data from scratch
393 or based on the estimated parameters of existing `momentuHMM` or `miSum` models. The
394 `simData` arguments are very similar to those used for model specification in `fitHMM`
395 (e.g., `dist`, `DM`) and data preparation in `prepData` (e.g., `spatialCovs`, `centers`),
396 but they include additional arguments, `lambda` and `errorEllipse`, for simulating
397 location data subject to temporal irregularity and measurement error, respectively.
398 The `spatialCovs` argument allows for rasters of spatio-temporal covariate values to
399 be utilized in simulation models, while the `centers` argument allows activity centers
400 to be incorporated. Thus `simData` can be used to simulate more ecologically-realistic
401 tracks (potentially subject to observation error) that can be useful for study design,
402 power analyses, and assessing model performance.

403 3 Examples

404 We will now demonstrate some of the capabilities of `momentuHMM` using real telemetry
405 data. These examples are intended for demonstration purposes only, and we do
406 not claim these example analyses represent improvements relative to previous or
407 alternative analyses for these data sets. While only some of the key workflow elements
408 are included here, complete R code and further details for these analyses can be found
409 in the online Supplementary Material.

410 3.1 African elephant

411 As our first example, we use an African elephant (*Loxodonta africana*) bull track
412 described in Wall *et al.* (2014) and publicly available from the movebank.org data

413 repository. In addition to hourly locations, the tag also collected external tempera-
 414 ture data. Location measurement error is negligible for these terrestrial GPS data,
 415 although about 1% of the hourly observations collected between 22 March 2008 and
 416 30 September 2010 are missing. Instead of simply ignoring these missing data, we
 417 can employ `crawlWrap` to predict the missing locations based on the CTCRW model
 418 of Johnson *et al.* (2008) prior to conducting our HMM analysis:

```
> crwOut <- crawlWrap(obsData = elephantData, timeStep = "hour",
+                      initial.state = inits, theta = c(4,-10), fixPar = c(NA,NA))
```

419 Here the desired time step is specified by the `timeStep` argument, and the `initial.state`,
 420 `theta`, and `fixPar` arguments are the same as for `crawl::crwMLE` (Johnson 2017).
 421 Now that we have our complete set of temporally-regular location data, we can begin
 422 fitting HMMs using `fithMM`.

423 During preliminary exploratory data analysis, autocorrelation function estimates
 424 suggested there are 24-hour cycles in the step length data (see Figure 1), and this
 425 presents an opportunity to demonstrate the use of the `cosinor` function for incorpo-
 426 rating cyclical behavior in model parameters using `momentuHMM`. But first we must
 427 create a `momentuHMMData` object and the 24-hour `cosinor` model covariate:

```
> ### create momentuHMMData object from crwData object
> dataHMM <- prepData(crwOut, covNames="temp")
> ### add cosinor covariate based on hour of day
> dataHMM$hour <- as.integer(strftime(dataHMM$time, format = "%H", tz="GMT"))
```

428 As seen here, the function `prepData` can also be used for pre-processing the best pre-
 429 dicted track data from `crawlWrap` output. The 24-hour `cosinor` covariate (“hour”)
 430 is simply a set of integers $(0, 1, \dots, 23)$ indicating the hour of day for each obser-
 431 vation. Now that the data have been pre-processed and the `cosinor` covariate has

432 been created, we're ready to fit a 2-state HMM to the elephant track that includes
 433 temperature effects on the turning angle concentration parameters and cycling tem-
 434 perature effects (with a 24-hour periodicity) on the step length and state transition
 435 probability parameters:

```
> DM <- list(step = list(mean = ~ temp * cosinor(hour, period = 24),
+                          sd = ~ temp * cosinor(hour, period = 24)),
+            angle = list(concentration = ~ temp))
> formula <- ~ temp * cosinor(hour, period = 24)
> elephantFit <- fitHMM(data = dataHMM, nbStates = 2,
+                       dist = list(step = "gamma", angle = "wrrpcauchy"),
+                       Par0 = Par0$Par, beta0 = Par0$beta,
+                       DM = DM, formula = formula,
+                       estAngleMean = list(angle=FALSE))
> plot(elephantFit, plotCI = TRUE, covs = data.frame(hour=12))
```

436 The function `getPar0` was used to extract parameters from an existing model and
 437 organize the starting values for the data stream (`Par0`) and state transition proba-
 438 bility (`beta0`) working parameters in the correct format based on `DM` and `formula`,
 439 respectively.

440 The above model identified an “encamped” (0.74 of time steps) and “exploratory”
 441 (0.26 of time steps) state (Figure 1), and was overwhelmingly supported by AIC
 442 when compared to alternative models with fewer covariates. Interestingly, this model
 443 suggests step lengths and directional persistence for the “encamped” state decreased
 444 as temperature increased, step lengths for both states tended to decrease in the
 445 late evening and early morning, and transition probabilities from the “encamped” to
 446 “exploratory” state decreased as temperature increased (Figure 2). Autocorrelation
 447 function plots of the pseudo-residuals (Figure 1) indicate this model explained much
 448 of the periodicity in step length, although there does still appear to be some room
 449 for improvement.

450 3.2 Northern fur seal

451 In our second example, we use the northern fur seal (*Callorhinus ursinus*) example
452 from McClintock *et al.* (2014) to demonstrate the use of additional data streams for
453 distinguishing behaviors with similar horizontal trajectories in a multivariate HMM.
454 The data consist of 241 temporally-irregular Fastloc GPS locations obtained during
455 a foraging trip of a nursing female near the Pribilof Islands of Alaska, USA, from
456 10-17 October 2007. The tag included time-depth recording capabilities, and the
457 dive activity data were summarized as the number of foraging dives over $T = 228$
458 temporally-regular 1 hr time steps. To fit the $Z = 3$ state (1="resting", 2="foraging",
459 3="transit") of McClintock *et al.* (2014) using `momentuHMM`, we first used `crawlWrap`
460 to predict temporally-regular locations at 1 hr time steps assuming a bivariate normal
461 measurement error model and merged the results with the foraging dive data using
462 the `crawlMerge` function (see Supplementary Material). Then multiple imputation
463 was used to account for location measurement error by repeatedly fitting the HMM
464 to `nSims` realizations of the position process using `MifitHMM`:

```
> nbStates <- 3
> stateNames <- c("resting", "foraging", "transit")
> dist <- list(step = "gamma", angle = "wrpcauchy", dive = "pois")
> Par0 <- getParDM(nbStates = nbStates, dist = dist,
+                 Par = Par, DM = DM, cons = cons,
+                 estAngleMean = list(angle = FALSE))
> fixPar <- list(dive = c(-100, NA, NA))
> nfsFits <- MifitHMM(crwOut, nSims = 100, nbStates = nbStates, dist = dist,
+                   Par0 = Par0, DM = DM, cons = cons,
+                   estAngleMean = list(angle = FALSE),
+                   fixPar = fixPar, retryFits = 30,
+                   stateNames=stateNames)
> plot(nfsFits)
```

465 Here we specified a gamma distribution for step length ('step'), wrapped Cauchy
 466 distribution for turning angle ('angle'), and Poisson distribution for the number of
 467 foraging dives ('dive'). The function `getParDM` was used to organize the starting
 468 values for the data stream working parameters (`Par0`) in the correct format based
 469 on `DM`, `cons`, and estimates of the natural parameters (`Par`) from McClintock *et al.*
 470 (2014). As detailed in Supplementary Material, the `DM` and `cons` arguments were
 471 specified to avoid label switching among the `nSims` imputed data model fits and
 472 enforce similar state-dependent probability distribution constraints as McClintock
 473 *et al.* (2014), e.g., constraining the Poisson rate parameters such that the "foraging"
 474 state tends to have higher numbers of foraging dives than the "transit" state ($\lambda_2 >$
 475 λ_3). To prohibit foraging dives for the "resting" state, we used the `fixPar` argument
 476 to effectively fix the Poisson rate parameter to zero on the natural scale (i.e. $\lambda_1 \approx$
 477 0). To help deal with the problem of convergence to local maxima, the `retryFits`
 478 argument allows users to specify the number of times to attempt to re-fit each model
 479 using random perturbations of the parameter estimates as the starting values for
 480 optimization.

481 The results are very similar to those of the discrete-time model of McClintock
 482 *et al.* (2014), with periods of foraging often followed by resting (Figure 3). The
 483 "activity budgets" (i.e. the proportion of time steps allocated to each state) calculated
 484 by `MIpool` based on the estimated state sequences for each imputation were 0.3 (95%
 485 CI: 0.22–0.39) for "resting", 0.29 (95% CI: 0.22–0.36) for "foraging", and 0.41 (95%
 486 CI: 0.32–0.52) for "transit".

3.3 Loggerhead turtle

For our third example, we demonstrate how to model movement direction and step length as a function of angular covariates using hitherto unpublished loggerhead turtle (*Caretta caretta*) data for a captive-raised juvenile released in 2012 on the coast of North Carolina, USA. The data consist of 165 temporally-irregular Argos locations subject to measurement error and rasters of daily ocean surface currents collected between 20 November and 19 December 2012. Assuming a gamma distribution for step length (l_t) and a wrapped Cauchy distribution for turning angle (ϕ_t), we model the mean step length parameter (μ_t^l) as a function of ocean surface current speed (w_t) and direction (r_t) relative to the bearing of movement (b_t):

$$\mu_t^l = \exp(\beta_0^l + \beta_1^l w_t \cos(b_t - r_t)), \quad (3)$$

and the turning angle mean parameter (μ_t^ϕ) as a trade-off between short-term directional persistence and bias in the direction of ocean surface currents using the circular-circular regression link function:

$$\mu_t^\phi = \text{atan2}(\sin(d_t)\beta^\phi, 1 + \cos(d_t)\beta^\phi), \quad (4)$$

where $d_t = \text{atan2}(\sin(r_t - b_{t-1}), \cos(r_t - b_{t-1}))$.

We wish to fit a 2-state HMM to the turtle data, with a “foraging” state unaffected by currents and a “transit” state potentially influenced by ocean surface currents as in Eqs. 3 and 4. We used `crawlWrap` to predict $T = 350$ temporally-regular locations at 2 hr time steps assuming a bivariate normal measurement error model that accounts for the Argos location quality class (i.e. 3,2,1,0,A,B) of each observation (see Supplementary Material). We then again used multiple imputation to account

507 for location uncertainty by repeatedly fitting the HMM to `nSims` realizations of the
 508 position process using `MifitHMM`. We first draw `nSims` realizations of the position
 509 process and extract the corresponding spatial covariates from the raster bricks for
 510 ocean surface current speed (“`speedBrick`”) and direction (“`dirBrick`”) using `MifitHMM`
 511 with `fit=FALSE`:

```
> miTurtleData <- MifitHMM(crwOut, nSims = 100, fit=FALSE,
+                           spatialCovs = list(w = speedBrick, d = dirBrick, r = dirBrick),
+                           angleCovs = "d")
```

512 When the `fit` argument is `FALSE`, `MifitHMM` returns a list of length `nSims` composed
 513 of `momentuHMMData` objects (`miData`). For convenience and ease of interpretation,
 514 we manually added an additional covariate ($angle_osc = \cos(b_t - r_t)$) to each of the
 515 imputed data sets and fitted the 2-state HMM using Eqs. 3 and 4 for state 2 (“tran-
 516 sit”):

```
> nbStates<-2
> dist <- list(step = "gamma", angle = "wrpcauchy")
> DM <- list(step = list(mean = ~state2(w:angle_osc), sd = ~1),
+            angle = list(mean = ~state2(d), concentration= ~1))
> turtleFits <- MifitHMM(miTurtleData$miData, nbStates = nbStates, dist = dist,
+                       Par0 = Par0, DM = DM,
+                       estAngleMean = list(angle = TRUE),
+                       circularAngleMean = list(angle = TRUE))
> plot(turtleFits, plotCI = TRUE, covs = data.frame(angle_osc = cos(0)))
```

517 Note that the `state2` special function in `DM` indicates the covariate formulas are
 518 specific to state 2 (“transit”) and the `circularAngleMean` argument indicates that
 519 circular-circular regression link function is to be used on the mean turning angle
 520 parameter as in Eq. 4.

521 For the “transit” state, pooled parameter estimates indicated step lengths in-
 522 creased with ocean surface current speed and as the bearing of movement aligned

523 with ocean surface current direction ($\beta_1^l = 0.43$, 95% CI: $0.1 - 0.77$; Figure 4). The
 524 estimated wrapped Cauchy distribution for turning angle had mean angles (μ_t^ϕ)
 525 biased towards the direction of ocean surface currents for each time step ($\beta^\phi =$
 526 0.24 , 95% CI: $-0.01 - 0.5$), with concentration parameter $\rho_2^\phi = 0.85$ (95% CI:
 527 $0.77-0.92$) indicating turning angles were concentrated at μ_t^ϕ . Thus movement
 528 during the “transit” state appears to strongly follow ocean surface currents (mean
 529 $angle_osc = 0.87$, $SD = 0.23$), while movement during the “foraging” state exhibited
 530 shorter step lengths ($\mu_1^l = 2996\text{m}$, 95% CI: $1791 - 4202$) perpendicular to ocean sur-
 531 face currents (mean $angle_osc = 0.07$, $SD = 0.26$), with no directional persistence
 532 ($\rho_1^\phi = 0.03$). The turtle spent 0.53 (95% CI: $0.34-0.71$) of the 2 hr time steps in
 533 the “foraging” state and 0.47 (95% CI: $0.29-0.66$) of time steps in the “transit” state
 534 as it travelled northeast along a predominant current until it (presumably) found an
 535 attractive foraging patch (Figure 4).

536 **3.4 Grey seal**

537 For our last example, we perform a similar analysis of a grey seal (*Halichoerus gry-*
 538 *pus*) track that was originally conducted by McClintock *et al.* (2012) using Bayesian
 539 methods and (computationally-intensive) Markov chain Monte Carlo. The data con-
 540 sist of 1045 temporally-irregular Fastloc GPS locations collected in the North Sea
 541 between 9 April and 11 August 2008. Because the seal repeatedly visited the same
 542 haulout and foraging locations, it provides a nice example for demonstrating how
 543 to implement biased movements relative to activity centers using **momentuHMM**. Mc-
 544 Clintock *et al.* (2012) fitted a 5-state model to these data that included three center
 545 of attraction states, with movement biased towards two haulout sites (“Abertay”

546 and “Farne Islands”) and a foraging area (“Dogger Bank”), and two “exploratory”
 547 states (“low speed”, “high speed”) that were unassociated with an activity center.
 548 After using `crawlWrap` to predict $T = 1515$ temporally-regular locations at 2 hr
 549 time steps including a bivariate normal measurement error model (see Supplemen-
 550 tary Material), we can perform a very similar analysis to McClintock *et al.* (2012)
 551 in `momentuHMM` by using the `centers` argument and state-specific functions for the
 552 probability distribution parameters. A cluster analysis on the observed locations
 553 using the R package `dtwclust` (Sarda-Espinosa 2017) identified three centroids with
 554 coordinates that were nearly identical to the three activity centers (“Abertay”, “Farne
 555 Islands”, and “Dogger Bank”) identified by McClintock *et al.* (2012). We use these
 556 coordinates derive covariates relative to the activity centers when drawing `nSims`
 557 realizations of the position process:

```

> crwSim <- MIfitHMM(crwOut, nSims = 100, fit=FALSE,
+                   center = centers)
  
```

558 Specifying the `centers` argument results in the calculation of two covariates for each
 559 activity center: the distance (with ‘.dist’ suffix) and angle (with ‘.angle’ suffix) from
 560 each location at time t . These covariates can then be used to model parameters as a
 561 function of the distance and angle to activity centers for each time step:

```

> dist <- list(step = "weibull", angle = "wrpcauchy")
> distFormula <- ~state1(I(Abertay.dist>2500)) + state2(I(Farne.dist>2500))
>               + state3(I(Dogger.dist>15000))
> angleFormula <- ~state1(Abertay.angle) + state2(Farne.angle)
>               + state3(Dogger.angle)
> stepDM <- list(shape = distFormula, scale = distFormula)
> angleDM <- list(mean = angleFormula, concentration = distFormula)
> DM <- list(step = stepDM, angle = angleDM)
  
```

562 Similar to McClintock *et al.* (2012), we assume a Weibull distribution for step length

563 where both the shape and scale parameter depend on the distance from the location
 564 at time t to each activity center. For the activity centers on land (“Abertay” and
 565 “Farne”), we allow the (state-dependent) step length parameters to change when the
 566 seal is beyond 2500m of the haulout. For the “Dogger” activity center, we allow the
 567 parameters to change when the seal is beyond 15000m of this (presumably) foraging
 568 area. We thus allow the movement behavior to change within these activity center
 569 states upon entering or leaving the vicinity of these sites. We assume a wrapped
 570 Cauchy distribution for turning angle with (state-dependent) mean angle derived
 571 from the direction to each activity center at time t , and the concentration parameter
 572 is modeled similarly to the step length parameters. For the two “exploratory” states,
 573 we assumed they are simple random walks unaffected by proximity to activity centers.
 574 To complete our model specification, we use the `knownStates` argument to assign
 575 the seal to the corresponding activity center state whenever it was within the 2500m
 576 (haulout area) or 15000m (foraging area) thresholds for each imputed data set:

```

> greySealFits <- MifitHMM(miDat, nSims = 400,
+                           nbStates = 5, dist = dist,
+                           Par0 = Par0, beta0 = beta0, fixPar = fixPar,
+                           formula = distFormula,
+                           estAngleMean = list(angle=TRUE),
+                           circularAngleMean = list(angle=TRUE),
+                           DM = DM, knownStates = knownStates)
> plot(greySealFits, plotCI = TRUE)

```

577 As with the step length and turning angle concentration parameters, the state transi-
 578 tion probabilities are also allowed to change as a function of distance to activity cen-
 579 ters (as specified by the `formula` argument). The starting values (`Par0` and `beta0`)
 580 for each imputation were extracted from a single HMM fitted to the best predicted
 581 locations from `crawlWrap`, and `fixPar` was used to remove short-term directional

582 persistence (and thus formulate the model as a mixture of biased and simple random
583 walks; see Supplementary Material for these finer details).

584 Estimated activity budgets for the 5 states of this multiple imputation HMM
585 were 0.28 (0.27 – 0.3) for the “Abertay” haul-out state, 0.12 (0.11 – 0.14) for the
586 “Farne Islands” haul-out state, 0.37 (0.35 – 0.38) for the “Dogger Bank” foraging
587 state, 0.09 (0.03 – 0.2) for a low-speed “exploratory” state, and 0.14 (0.08 – 0.23) for
588 a high-speed “exploratory” state. All three activity center states exhibited shorter
589 step lengths and less biased movements when within the vicinity of these targets
590 (Figure 5). Results from this analysis were thus very similar to those of McClintock
591 *et al.* (2012), but this implementation required far less computation time and no
592 custom model-fitting algorithms.

593 The `simData` function can be used to simulate tracks from a fitted model:

```
> greySealSim<-simData(model = greySealFits, centers = centers,  
+                       initialPosition = centers[1,],  
+                       obsPerAnimal = 1515)
```

594 A simulated track is presented along with the fitted track in Figure 6. While poten-
595 tially useful for study design, power analysis, and prediction, the `simData` function
596 can also be helpful in assessing goodness of fit by repeatedly drawing simulated data
597 sets from a fitted model and comparing them to observed properties of the data (e.g.
598 Morales *et al.* 2004).

599 4 Discussion

600 Here we have introduced version 1.0.0 of the R package `momentuHMM` and demon-
601 strated some of its capabilities for conducting multivariate HMM analyses with ani-
602 mal location, auxiliary biotelemetry, and environmental data. The package allows for

fitting (and simulating from) a suite of biased and correlated random walk movement
 process models (e.g. McClintock *et al.* 2012), can be used for an unlimited number of
 data streams and latent behavior states, includes multiple imputation methods to ac-
 count for measurement error, temporal irregularity, and other forms of missing data
 that would otherwise be prohibitive to maximum likelihood analysis, and integrates
 seamlessly with rasters to facilitate spatio-temporal covariate modelling. Because
 the package incorporates biased random walks, it can also be used to implement
 group dynamic models (e.g. Langrock *et al.* 2014). The package therefore greatly
 expands on available software and facilitates the incorporation of more ecological and
 behavioral realism for hypothesis-driven analyses of animal movement that account
 for many of the challenges commonly associated with telemetry data. While many of
 the features of **momentuHMM** were motivated by animal movement data, we note that
 the package is not limited to location data and can be used for analyzing any type
 of data that is amenable to (multivariate) HMMs.

Model fitting in **momentuHMM** is relatively fast because the forward algorithm (Eq.
 1) is coded in C++. Because multiple imputations are completely parallelizable,
 with sufficient processing power computation times for analyses that account for
 measurement error, temporal irregularity, or other forms of missing data need not
 be longer than that required to fit a single HMM. However, computation times will
 necessarily be slower as the number of states and/or parameters increase. For exam-
 ple, **momentuHMM** required about 1 hr to fit a single HMM with $N = 6$ states, seven
 data streams, and $T = 7414$ time steps (McClintock 2017).

As in any maximum likelihood analysis based on numerical optimization, com-
 putation times will also depend on the starting values (**Par0** and **beta0**). Specifying

627 “good” starting values is arguably the most challenging aspect of model fitting in
628 `momentuHMM`, particularly for the working scale coefficients when using covariates.
629 The `getPar`, `getPar0`, and `getParDM` functions are designed to help with the speci-
630 fication of starting values, and the `retryFits` argument in `crawlWrap`, `fitHMM`, and
631 `MIfitHMM` will re-optimize based on random perturbations of the parameters to help
632 explore the likelihood surface and diagnose convergence to local maxima. Optimiza-
633 tion for the circular-linear regression link function (`tan(mean/2)`; see Table 2) in
634 particular can be prone to local minima, so users are encouraged to explore a range
635 of starting values when fitting these models.

636 While `momentuHMM` includes functions for drawing realizations of the position pro-
637 cess based on the CTCRW model of Johnson *et al.* (2008), this is but one of many
638 methods for performing the first stage of multiple imputation. Realizations of the
639 position process from any movement model that accounts for measurement error
640 and/or temporal irregularity (e.g. Calabrese *et al.* 2016; Gurarie *et al.* 2017) could
641 be passed to `MIfitHMM` for HMM-type analyses in the second stage. Multiple im-
642 putation methods also need not be limited to these telemetry error scenarios. For
643 example, conventional missing data could also be imputed using standard techniques
644 (Rubin & Schenker 1986), thereby allowing the investigation of non-random mecha-
645 nisms for missingness that can be problematic if left unaccounted for in HMMs.

646 There remain many potential avenues for refining and extending the capabilities
647 of `momentuHMM`. Computation times could likely be improved by further optimizing
648 the R and C++ code for speed. Notable extensions include hidden semi-Markov
649 models and random effects on data stream probability distribution and state transi-
650 tion probability parameters (Zucchini *et al.* 2016). We would also like to incorporate

651 additional parameters for change-point thresholds and the locations of activity cen-
652 ters instead of requiring that they be pre-specified (and potentially compared using
653 AIC or other model selection criteria) as in grey seal example. Lastly, it is relatively
654 straightforward to add additional probability distributions, and we are pleased to do
655 so upon request. Practitioners interested in additional features for `momentuHMM` are
656 encouraged to contact the authors.

657 **Acknowledgments**

658 The findings and conclusions in the manuscript are those of the author(s) and do
659 not necessarily represent the views of the National Marine Fisheries Service, NOAA.
660 Any use of trade, product, or firm names does not imply an endorsement by the US
661 Government.

662 **References**

- 663 Beyer, H.L., Morales, J.M., Murray, D. & Fortin, M.J. (2013) The effectiveness
664 of bayesian state-space models for estimating behavioural states from movement
665 paths. *Methods in Ecology and Evolution*, **4**, 433–441.
- 666 Calabrese, J.M., Fleming, C.H. & Gurarie, E. (2016) ctmm: an R package for ana-
667 lyzing animal relocation data as a continuous-time stochastic process. *Methods in*
668 *Ecology and Evolution*, **7**, 1124–1132.
- 669 Cornelissen, G. (2014) Cosinor-based rhythmometry. *Theoretical Biology and Medical*
670 *Modelling*, **11**, 16.
- 671 Costa, D.P., Robinson, P.W., Arnould, J.P., Harrison, A.L., Simmons, S.E., Hassrick,
672 J.L., Hoskins, A.J., Kirkman, S.P., Oosthuizen, H., Villegas-Amtmann, S. *et al.*

673 (2010) Accuracy of argos locations of pinnipeds at-sea estimated using fastloc gps.
 674 *PloS one*, **5**, e8677.

675 DeRuiter, S.L., Langrock, R., Skirbutas, T., Goldbogen, J.A., Chalambokidis, J.,
 676 Friedlaender, A.S. & Southall, B.L. (2017) A multivariate mixed hidden markov
 677 model to analyze blue whale diving behaviour during controlled sound exposures.
 678 *The Annals of Applied Statistics*, **11**, 362–392.

679 Duchesne, T., Fortin, D. & Rivest, L.P. (2015) Equivalence between step selection
 680 functions and biased correlated random walks for statistical inference on animal
 681 movement. *PloS one*, **10**, e0122947.

682 Gilbert, P. & Varadhan, R. (2016) *numDeriv: Accurate Numerical Derivatives*. R
 683 package version 2016.8-1.

684 Gurarie, E., Fleming, C.H., Fagan, W.F., Laidre, K.L., Hernández-Pliego, J. &
 685 Ovaskainen, O. (2017) Correlated velocity models as a fundamental unit of an-
 686 imal movement: synthesis and applications. *Movement Ecology*, **5**, 13.

687 Hijmans, R.J. (2016) *raster: Geographic Data Analysis and Modeling*. R package
 688 version 2.5-8.

689 Hooten, M.B., Johnson, D.S., McClintock, B.T. & Morales, J.M. (2017) *Animal*
 690 *Movement: Statistical Models for Telemetry Data*. CRC Press.

691 Johnson, D.S. (2017) *crawl: Fit Continuous-Time Correlated Random Walk Models*
 692 *to Animal Movement Data*. R package version 2.1.2.

- 693 Johnson, D.S., London, J.M., Lea, M.A. & Durban, J.W. (2008) Continuous-time
694 correlated random walk model for animal telemetry data. *Ecology*, **89**, 1208–1215.
- 695 Jonsen, I.D., Flemming, J.M. & Myers, R.A. (2005) Robust state–space modeling of
696 animal movement data. *Ecology*, **86**, 2874–2880.
- 697 Langrock, R., Hopcraft, G., Blackwell, P., Goodall, V., King, R., Niu, M., Patterson,
698 T., Pedersen, M., Skarin, A. & Schick, R. (2014) Modelling group dynamic animal
699 movement. *Methods in Ecology and Evolution*, **5**, 190–199.
- 700 Langrock, R., King, R., Matthiopoulos, J., Thomas, L., Fortin, D. & Morales, J.M.
701 (2012) Flexible and practical modeling of animal telemetry data: hidden markov
702 models and extensions. *Ecology*, **93**, 2336–2342.
- 703 McClintock, B.T. (2017) Incorporating telemetry error into hidden markov models of
704 animal movement using multiple imputation. *Journal of Agricultural, Biological,*
705 *and Environmental Statistics*, **in press**.
- 706 McClintock, B.T., Johnson, D.S., Hooten, M.B., Ver Hoef, J.M. & Morales, J.M.
707 (2014) When to be discrete: the importance of time formulation in understanding
708 animal movement. *Movement Ecology*, **2**, 21.
- 709 McClintock, B.T., King, R., Thomas, L., Matthiopoulos, J., McConnell, B.J. &
710 Morales, J.M. (2012) A general discrete-time modeling framework for animal move-
711 ment using multistate random walks. *Ecological Monographs*, **82**, 335–349.
- 712 McClintock, B.T., London, J.M., Cameron, M.F. & Boveng, P.L. (2017) Bridging the
713 gaps in animal movement: hidden behaviors and ecological relationships revealed
714 by integrated data streams. *Ecosphere*, **8**, e01751.

- 715 McClintock, B.T., Russell, D.J., Matthiopoulos, J. & King, R. (2013) Combin-
 716 ing individual animal movement and ancillary biotelemetry data to investigate
 717 population-level activity budgets. *Ecology*, **94**, 838–849.
- 718 Michelot, T., Langrock, R. & Patterson, T.A. (2016) moveHMM: An R package for
 719 the statistical modelling of animal movement data using hidden markov models.
 720 *Methods in Ecology and Evolution*, **7**, 1308–1315.
- 721 Morales, J.M., Haydon, D.T., Frair, J., Holsinger, K.E. & Fryxell, J.M. (2004) Ex-
 722 tracting more out of relocation data: building movement models as mixtures of
 723 random walks. *Ecology*, **85**, 2436–2445.
- 724 R Core Team (2016) *R: A Language and Environment for Statistical Computing*. R
 725 Foundation for Statistical Computing, Vienna, Austria.
- 726 Rubin, D.B. & Schenker, N. (1986) Multiple imputation for interval estimation from
 727 simple random samples with ignorable nonresponse. *Journal of the American*
 728 *Statistical Association*, **81**, 366–374.
- 729 Sarda-Espinosa, A. (2017) *dtwclust: Time Series Clustering Along with Optimiza-*
 730 *tions for the Dynamic Time Warping Distance*. R package version 3.1.2.
- 731 Wall, J., Wittemyer, G., LeMay, V., Douglas-Hamilton, I. & Klinkenberg, B. (2014)
 732 Elliptical time-density model to estimate wildlife utilization distributions. *Methods*
 733 *in Ecology and Evolution*, **5**, 780–790.
- 734 Whoriskey, K., Auger-Méthé, M., Albertsen, C.M., Whoriskey, F.G., Binder, T.R.,
 735 Krueger, C.C. & Mills Flemming, J. (2017) A hidden markov movement model for

736 rapidly identifying behavioral states from animal tracks. *Ecology and Evolution*,
737 **7**, 2112–2121.

738 Zucchini, W., MacDonald, I.L. & Langrock, R. (2016) *Hidden Markov Models for*
739 *Time Series: An Introduction Using R*. CRC Press.

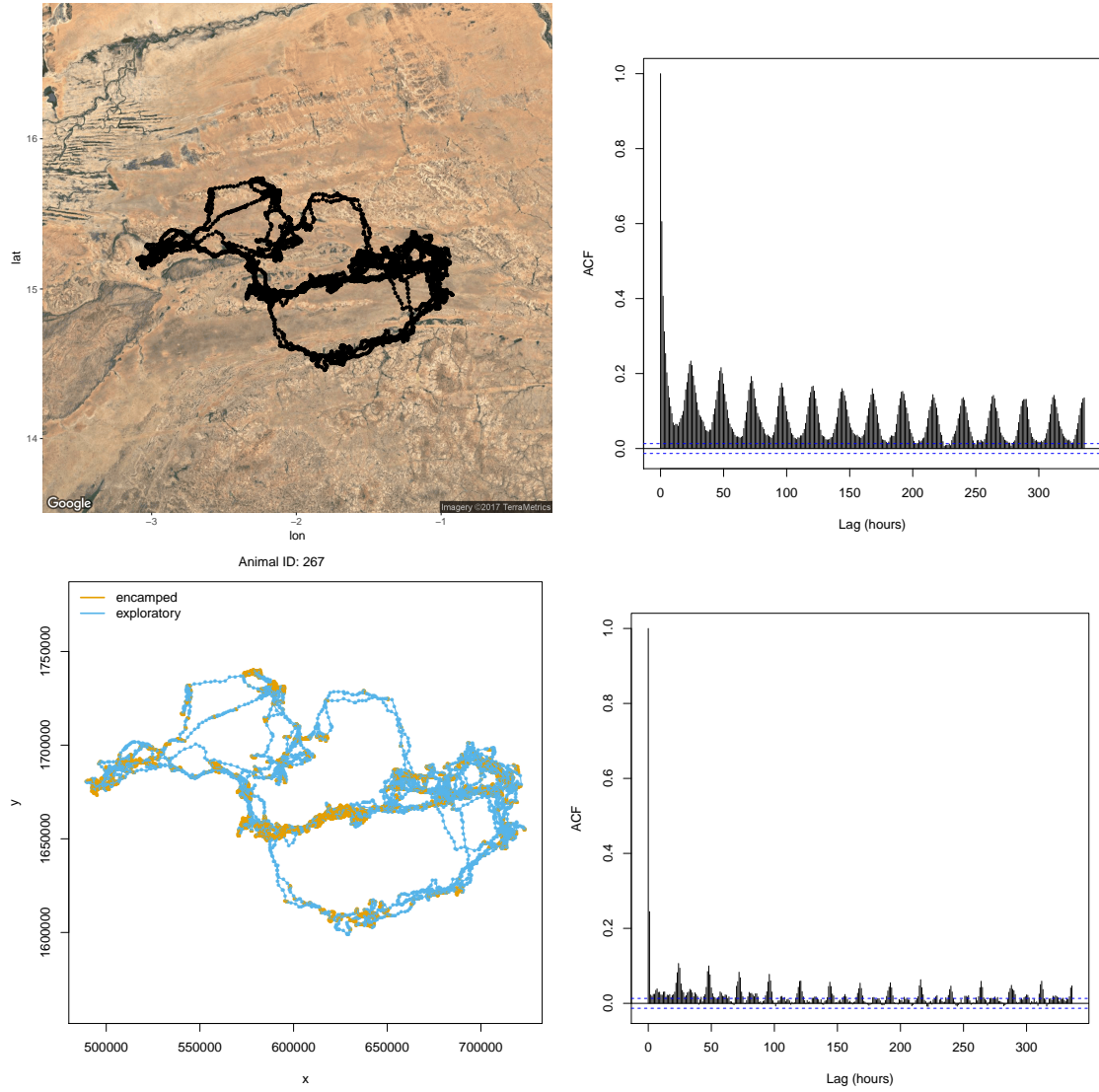


Figure 1. Plot of the elephant track produced using the ‘plotSat’ function (top-left panel), autocorrelation function (ACF) plot of the corresponding step length data (top-right panel), plot of the Viterbi-decoded state sequence for the 2-state (“encamped” and “exploratory”) model generated using the generic ‘plot’ function (bottom-left panel), and the step length pseudo-residual ACF plot for this model using the ‘plotPR’ function (bottom-right panel).

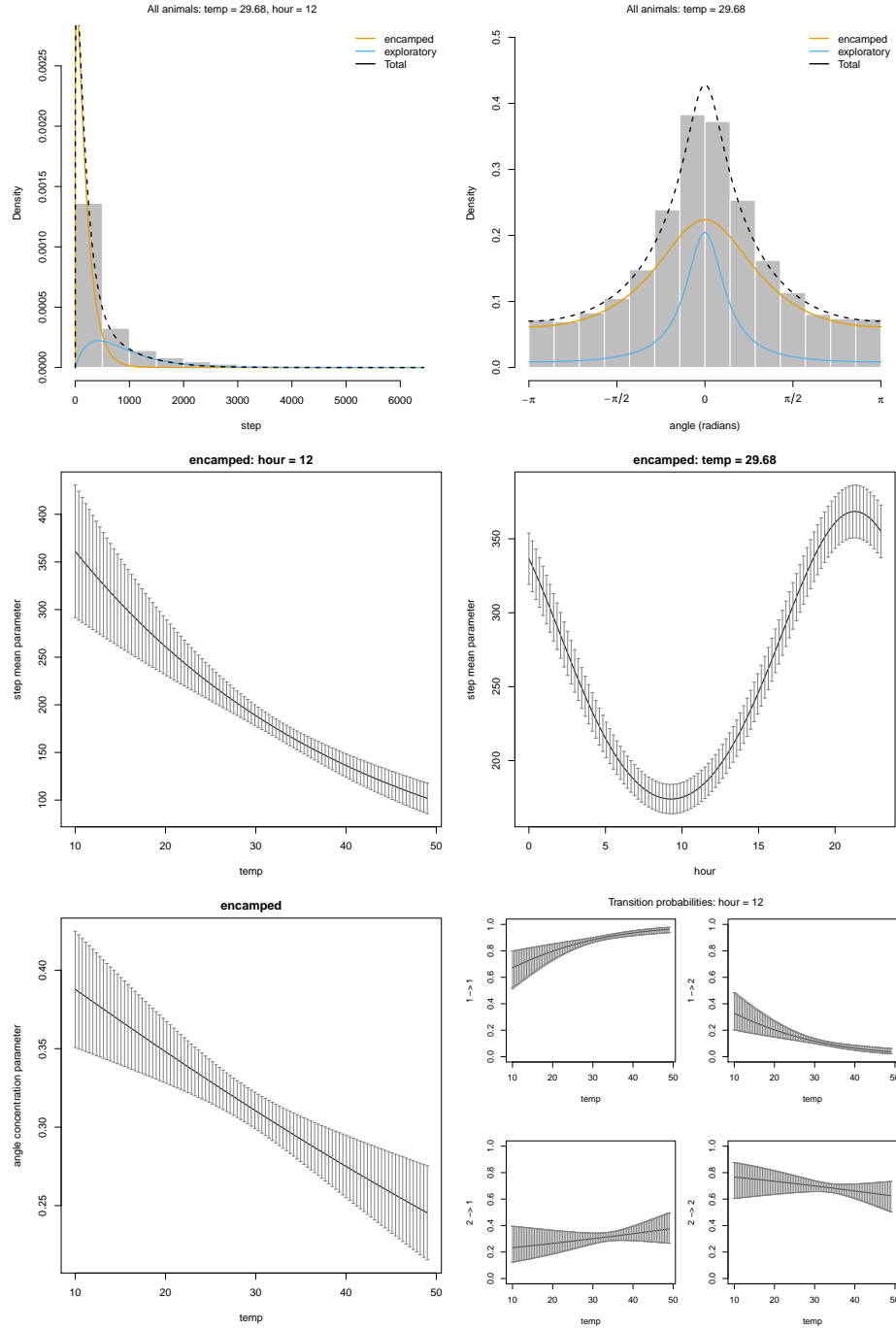


Figure 2. Selected plots for the 2-state (“encamped” and “exploratory”) African elephant example generated using the generic ‘plot’ function. Top panels present histograms of the step length (top-left) and turning angle (top-right) data along with the estimated state-dependent probability distributions based on the mean temperature (temp = 29.7 degrees celsius) at 12:00 GMT (hour = 12). Middle panels present estimates (and 95% confidence intervals) for the step length mean parameter of the “encamped” state as a function of temperature and hour of day. Bottom-left panel presents estimates for the turning angle concentration parameter of the “encamped” state as a function of temperature. Bottom-right panel presents estimated state transition probabilities (1 = “encamped”, 2 = “exploratory”) as a function of temperature at 12:00 GMT.

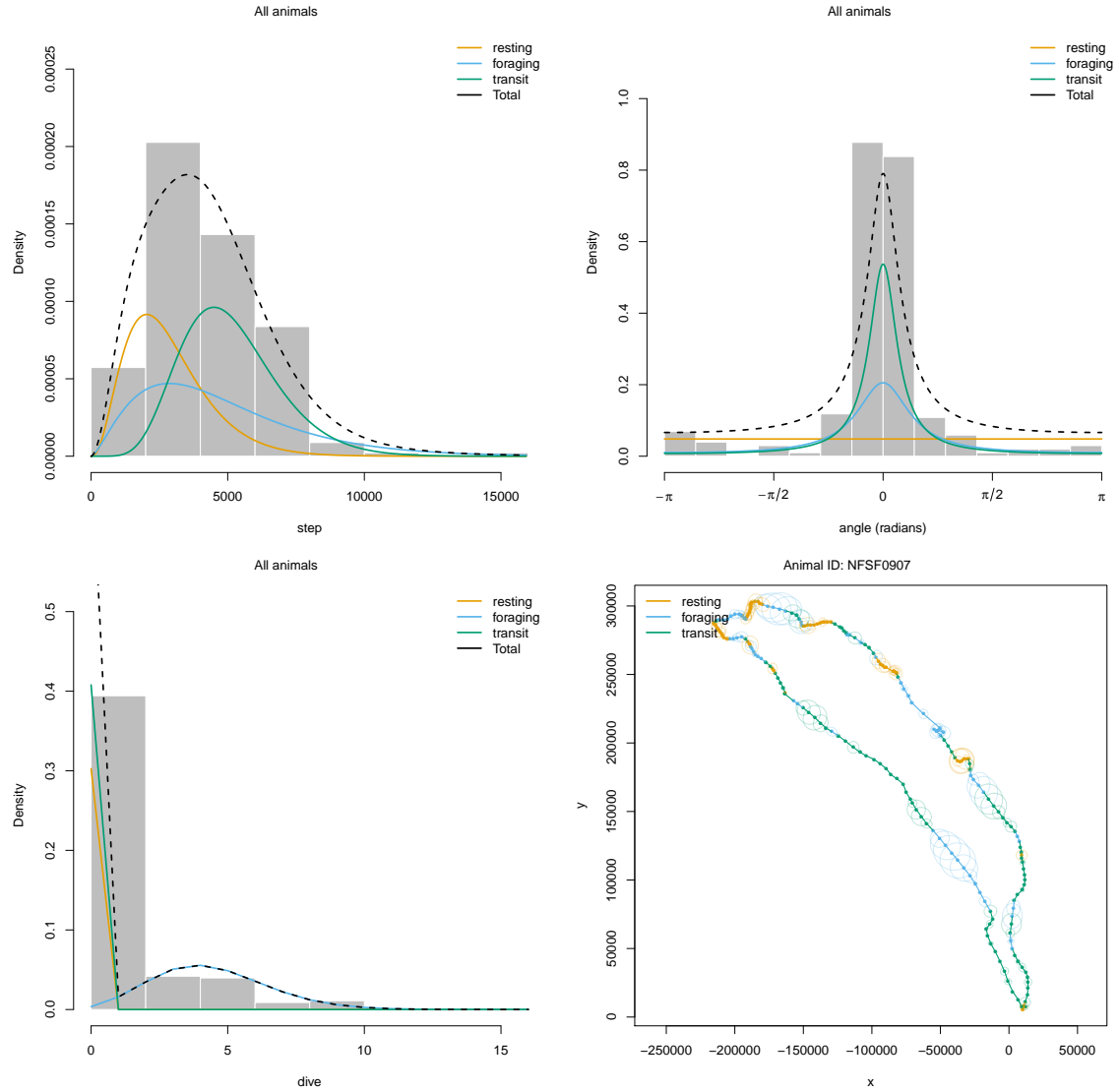


Figure 3. Plots of the northern fur seal example results generated using the generic ‘plot’ function. The estimated probability distributions for step length (top-left panel), turning angle (top-right panel), and number of foraging dives (bottom-left panel) for the 3-state (“resting”, “foraging”, and “transit”) model are plotted along with histograms of these data streams. The temporally-regular predicted locations (and 95% ellipsoidal confidence bands) and estimated states are plotted in the bottom-right panel. All estimates are pooled across multiple imputations of the position process and thus reflect uncertainty attributable to location measurement error and temporally-irregular observations.

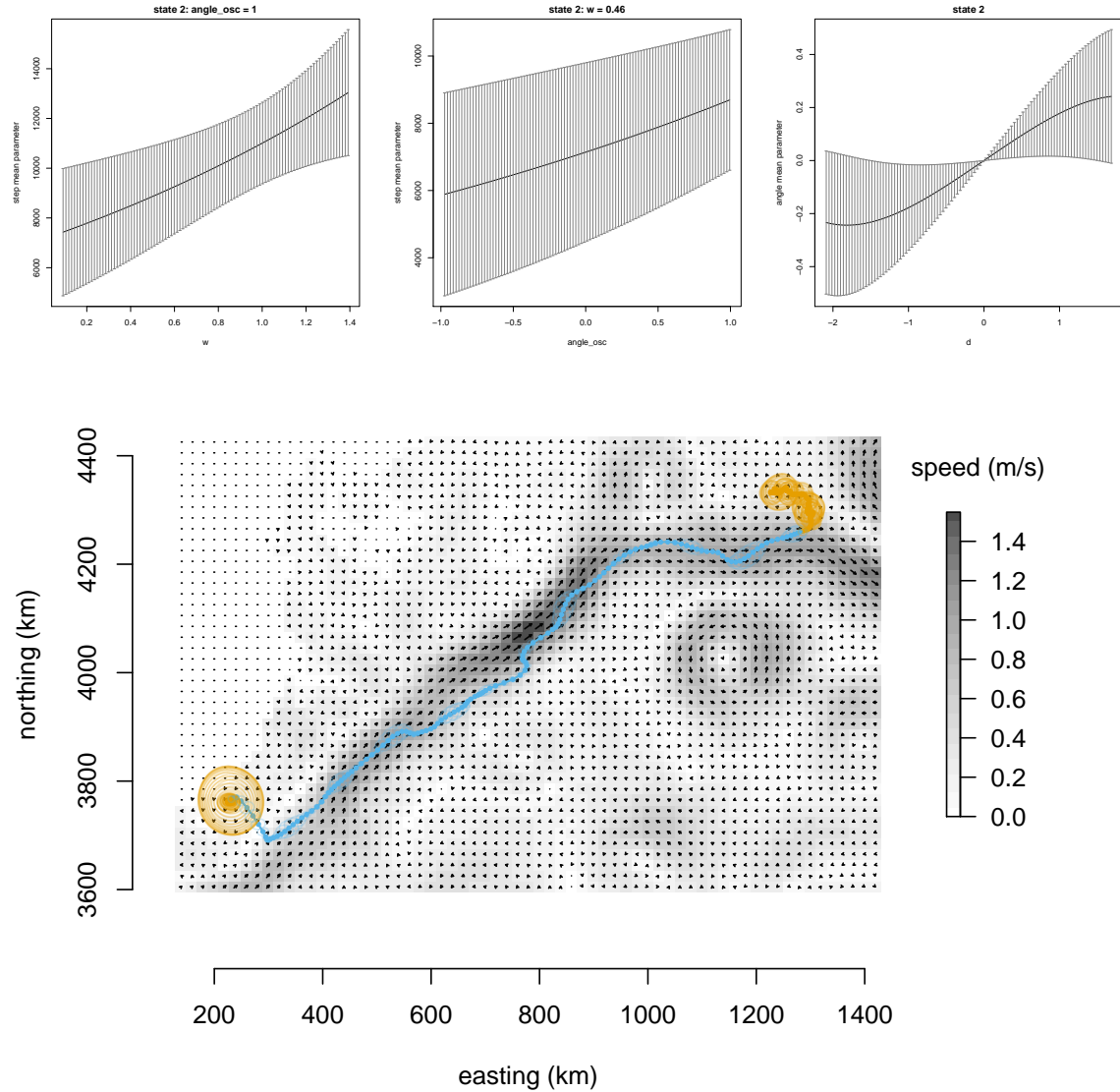


Figure 4. Selected results from the loggerhead turtle example. Top panels include estimates and 95% confidence intervals for the mean step length parameter as a function of ocean surface current speed (w) when ocean surface current direction (r_t) is the same as the bearing (b_t) of movement (i.e. $angle_osc = \cos(b_t - r_t) = 1$; top-left panel), mean step length parameter as a function of $angle_osc$ at the mean ocean surface current speed ($w = 0.46$ m/s; top-middle panel), and mean turning angle parameter as a function of $d_t = \text{atan2}(\sin(r_t - b_{t-1}), \cos(r_t - b_{t-1}))$ (top-right panel). Bottom panel plots the pooled track, 95% error ellipse confidence bands, and state (orange = “foraging”, blue = “transit”) estimates based on multiple imputations of the position process relative to ocean surface current speed (m/s) and direction on 2 December 2012.

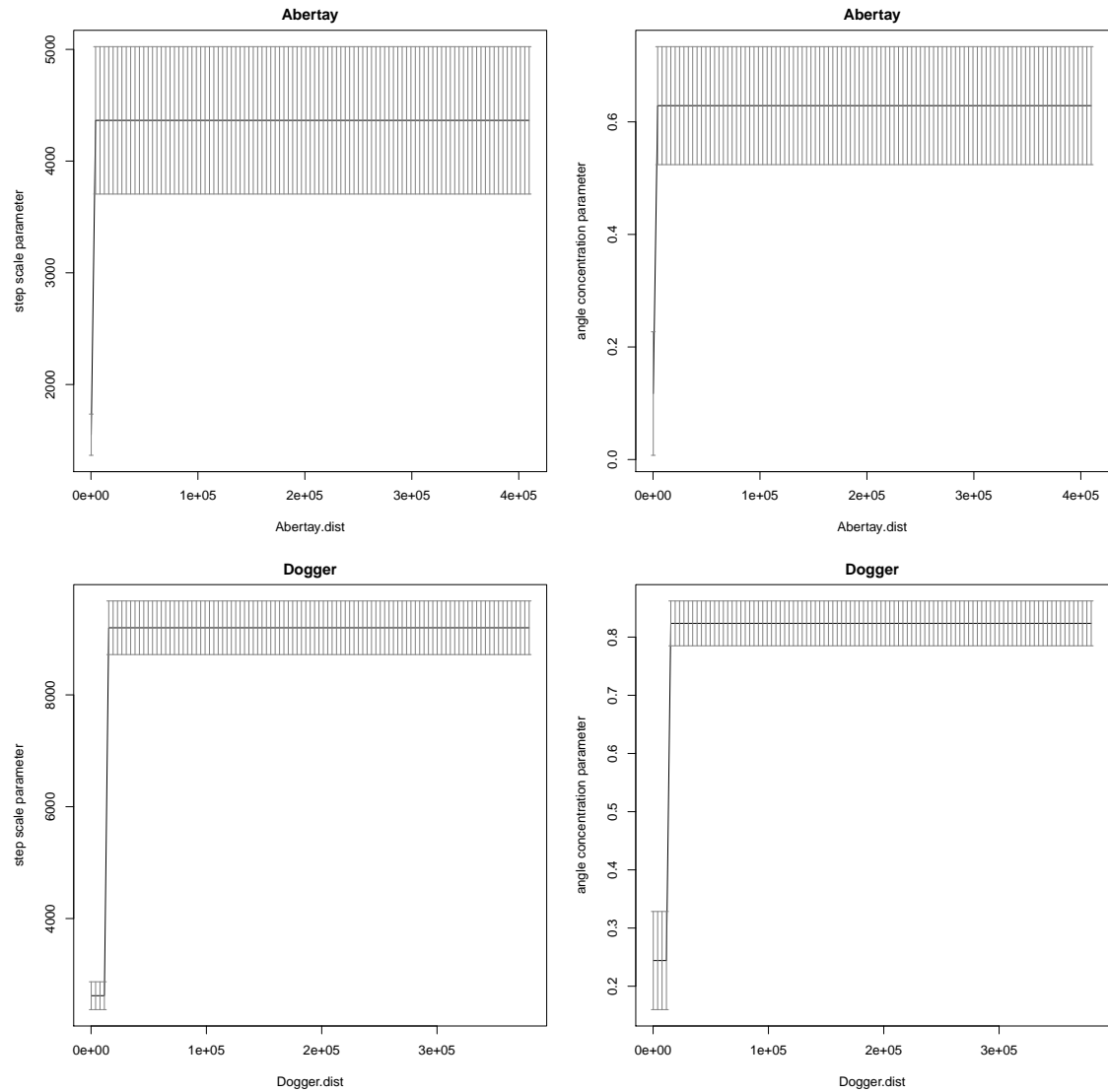


Figure 5. Selected results from the grey seal example. Panels include estimates and 95% confidence intervals for the “Abertay” haul-out state step length scale parameter as a function of distance in meters (‘Abertay.dist’; top-left panel), “Abertay” haul-out state turning angle concentration parameter as a function of distance (top-right panel), “Dogger Bank” foraging state step length scale parameter as a function of distance (‘Dogger.dist’; bottom-left panel), and the “Dogger Bank” foraging state turning angle concentration parameter as a function of distance (bottom-right panel).

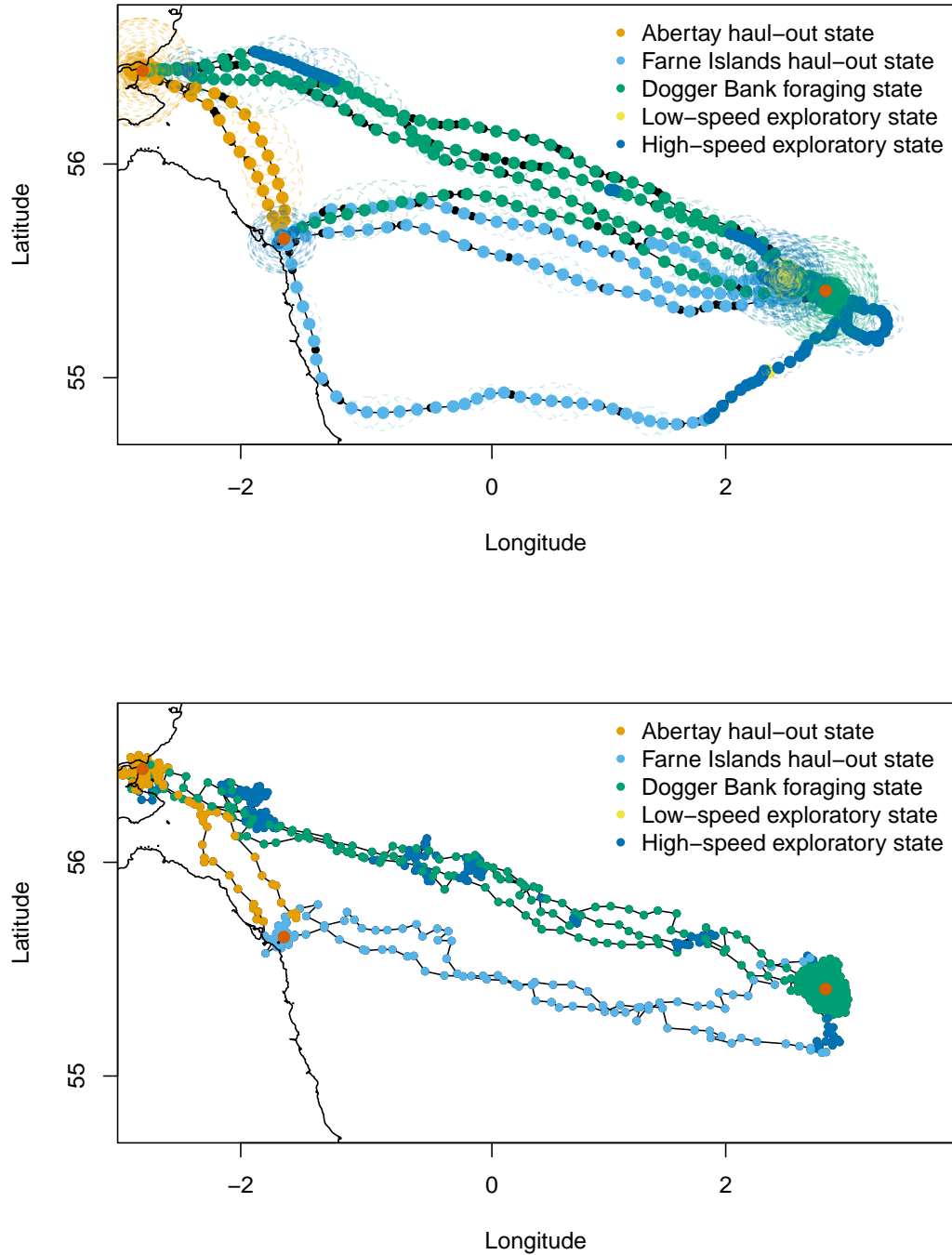


Figure 6. Fitted and simulated tracks from the grey seal example. This seal tended to move in a clockwise fashion between two haul-out locations (“Abertay” and “Farne Islands”) and a foraging area (“Dogger Bank”) in the North Sea. Top panel plots the pooled track, 95% error ellipse confidence bands, and state estimates based on the 5-state HMM fitted to multiple imputations of the position process. Red points indicate the locations of the three activity centers. Black points indicate the (temporally-irregular) observed locations. Bottom panel presents the locations and states for a track simulated from the fitted model using the ‘simData’ function.