

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 the $N \times N$ transition probability matrix $\boldsymbol{\Gamma}^{(t)} = \left(\gamma_{ij}^{(t)} \right)$, $\mathbf{P}(\mathbf{z}_t) = \text{diag}(p_1(\mathbf{z}_t), \dots, p_N(\mathbf{z}_t))$, $p_s(\mathbf{z}_t)$ 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

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

95 When data streams are observed without error and at regular time intervals, a
96 major advantage of HMMs is the relatively fast and efficient maximization of the
97 likelihood using the forward algorithm (Eq. 1). However, location measurement
98 error is rarely non-existent in animal-borne telemetry studies and depends on both
99 the device and the system under study. For example, GPS errors are typically less
100 than 50m, but Argos errors can exceed 10km (e.g. Costa *et al.* 2010). An extreme
101 case of missing data can arise when location data are obtained with little or no tem-
102 poral regularity, as in many marine mammal telemetry studies (e.g. Jonsen *et al.*
103 2005), such that few (if any) observations align with the regular time steps required
104 by discrete-time HMMs. When explicitly accounting for uncertainty attributable
105 to location measurement error, temporally-irregular observations, or other forms of
106 missing data, one must typically fit (multivariate) HMMs using computationally-
107 intensive (and often time-consuming) model fitting techniques such as Markov chain
108 Monte Carlo (Jonsen *et al.* 2005; McClintock *et al.* 2012). However, complex analy-
109 ses requiring novel statistical methods and custom model-fitting algorithms are not
110 practical for many practitioners.

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

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

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

141 probability distribution parameters using linear model formulas familiar to most R
142 users; 4) multiple imputation methods that account for observation error attributable
143 to measurement error and temporally-irregular or missing data (Hooten *et al.* 2017;
144 McClintock 2017); 5) seamless integration of spatio-temporal environmental covari-
145 ate data (e.g., wind direction, forest cover, sea ice concentration) using the **raster**
146 package (Hijmans 2016); 6) incorporation of “activity center” effects (e.g., areas asso-
147 ciated with attractive or repulsive forces); 7) circular-circular regression models for
148 angular probability distributions (Duchesne *et al.* 2015); 8) cosinor (e.g. Cornelissen
149 2014) and spline regression formulas for cyclical and other complicated behavioral
150 patterns; and 9) data simulation capabilities for power analyses and assessing model
151 performance, including simulation of location data subject to temporal irregularity
152 and/or measurement error.

153 In the following tutorial, we demonstrate some of the capabilities of **momentuHMM**
154 using real-world examples, including an example of periodic cycles in African elephant
155 movement, a 3-state (“resting”, “foraging”, “transit”) northern fur seal example in-
156 corporating auxiliary dive activity data (McClintock *et al.* 2014), a loggerhead turtle
157 example relating “foraging” and “transit” movements to ocean surface currents, and
158 a 5-state grey seal example incorporating biased movements toward haul-out and
159 foraging locations (McClintock *et al.* 2012). This brief tutorial includes workflows
160 for data formatting, model specification, model fitting, and diagnostics. While many
161 of the features of **momentuHMM** were motivated by animal movement data, the pack-
162 age can be used for analyzing any type of data that is amenable to (multivariate)
163 HMMs. Additional information, including help files, data, examples, and package us-
164 age is available by downloading the **momentuHMM** package from CRAN (<http://cran.r->

165 project.org) or github (<https://github.com/bmcclintock/momentuHMM>). This arti-
166 cle describes `momentuHMM` version 1.0.0.

167 2 `momentuHMM` overview

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

178 One of the key features of `momentuHMM` is the ability to specify an unlimited
179 number of HMM data streams from a broad range of commonly used probability
180 distributions. Any of the probability distribution parameters (as well as the state
181 transition probabilities) can be modelled as a function of environmental and indi-
182 vidual covariates using link functions (Table 2). For any given “natural scale” (or
183 “real scale”) probability distribution parameter θ , all of the link functions (g) in
184 `momentuHMM` are of the general form $g(\theta) = \mathbf{X}_\theta \boldsymbol{\beta}_\theta$, where \mathbf{X}_θ is the $T \times k$ design
185 matrix (composed of k covariates) and $\boldsymbol{\beta}_\theta$ is the corresponding k -vector of “working
186 scale” (or “beta scale”) parameters for θ . For example, suppose step length is as-
187 sumed to have a Gamma distribution, $l_t \mid S_t = s \sim \text{Gamma}(\mu_s, \sigma_s)$. In `momentuHMM`,

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)

the natural scale parameters for the Gamma distribution are the (state-dependent) step length mean ($\mu_s > 0$) and standard deviation ($\sigma_s > 0$). Because both of these parameters must be positive, the log link function is a natural choice for modelling these parameters as a function of covariates, e.g., $\log(\boldsymbol{\mu}) = \mathbf{X}_\mu \boldsymbol{\beta}_\mu$.

2.1 Data preparation and visualization

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

- **Data** A data frame with $T + 1$ rows including optionally a field ‘ID’ (identifiers for different individuals), coordinates from which step length (‘step’) and turning angle (‘angle’) data streams are to be calculated, any additional data streams, and any covariates identified in the `covNames` and `angleCovs` arguments. Alternatively, **Data** can be a `crwData` object returned by `crawlWrap`.
- **type** Coordinate type; ‘UTM’ if easting-northing or ‘LL’ if longitude-latitude.
- **coordNames** Names of the two coordinate columns in **Data**. If `coordNames=NULL` then step lengths, turning angles, and any location-based covariates (i.e., those specified by `spatialCovs`, `centers`, and `angleCovs`) are not calculated.
- **covNames** Character vector indicating the names of any covariates in **Data**. Any variables in **Data** (other than “ID”) that are not identified in `covNames` or `angleCovs` are assumed to be data streams.
- **spatialCovs** List of Raster-class objects (Hijmans 2016) containing spatio-temporally referenced covariates. Covariates specified by `spatialCovs` are ex-

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

210 tracted from the raster layer(s) based on the location data. Raster stacks may
211 also be included, in which case the appropriate z values (e.g. time, date) must
212 also be included in `Data`.

213 • `centers` 2-column matrix providing the coordinates for any activity centers
214 (e.g., potential centers of attraction or repulsion) from which distance and
215 angle covariates will be calculated based on the location data and returned in
216 the `momentuHMMData` object.

217 • `angleCovs` Character vector indicating the names of any circular-circular re-
218 gression angular covariates in `Data` or `spatialCovs` that need conversion from
219 standard direction (in radians relative to the x-axis) to turning angle (relative
220 to previous movement direction).

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

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

228 **2.2 HMM specification and fitting**

229 Once a `momentuHMMData` object has been created using `prepData`, then the data are
230 ready to be passed to the generalized multivariate HMM-fitting function `fitHMM`.
231 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 of 15 “wild haggis” tracks described in (Michelot *et al.* 2016) would be implemented 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))

```

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

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

any other value) using the `fixPar` argument, which can be useful for incorporating more behavioral realism. For example, `fixPar` can be used to prohibit or enforce switching from one particular state to another (possibly as a function of spatio-temporal covariates).

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

Note that when `DM` is specified for a data stream, the initial parameter values (`Par0`) for that data stream now correspond to columns of the resulting design matrix and must be on the working scale instead of the natural scale. In this case, because the log link is used for the natural parameters of the Gamma distribution, `Par0$step` was specified on the log scale. The functions `getPar0` and `getParDM` are designed to assist users in the specification of design matrices and corresponding initial values on the working scale for any given model (see package documentation for further details). `DM` formulas are just as flexible as the `formula` argument and, in addition

286 to common linear model formula functions and operators, can also include cyclical
 287 cosinor models, splines, factor variables, and state-specific probability distribution
 288 parameter formulas (see examples in sections 3.3 and 3.4). As with the state transi-
 289 tion probabilities, working parameters for probability distributions can also be fixed
 290 to user-specified values using the `fixPar` argument.

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

298 (note that column and rownames for pseudo-design matrices are not required but
 299 can be useful). Pseudo-design matrices allow for the sharing of common working
 300 parameters (such as intercept terms) among natural scale parameters, and this can be

301 used to constrain natural scale parameters (e.g., $\mu_1 \leq \mu_2$) when used in tandem with
 302 the `cons` argument. This is particularly useful for preventing state label switching
 303 when repeatedly fitting the same HMM using multiple imputation methods (see
 304 section 2.3).

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

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

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

323 can be addressed using circular-circular regression on movement direction, including
324 the effects of wind, sea surface currents (see example in section 3.3), and centers of
325 attraction or repulsion (see example in section 3.4)

326 **2.3 Multiple imputation**

327 When location data are temporally-irregular or subject to measurement error, then
328 they are not suitable for standard maximum-likelihood HMM analyses based on the
329 forward algorithm (Eq. 1). In this case, `momentuHMM` can be used to perform the
330 2-stage multiple imputation approach of McClintock (2017). The basic concept is
331 to first employ a single-state (i.e., $N = 1$) movement model that is relatively easy
332 to fit but can accommodate location measurement error and temporally-irregular or
333 missing observations (e.g. Johnson *et al.* 2008). The second stage involves repeatedly
334 fitting the desired HMM to m temporally-regular realizations of the position process
335 drawn from the model output of the first stage. Data streams or covariates that are
336 dependent on location (e.g., step length, turning angle, habitat type, snow depth,
337 sea surface temperature) will of course vary among the m realizations of the position
338 process, and the pooled inferences across the HMM analyses therefore reflect location
339 uncertainty.

340 There are three primary functions (`MifitHMM`, `MIpool`, and `crawlWrap`) for per-
341 forming multiple imputation HMM analyses in `momentuHMM`, and all rely on parallel
342 processing to speed up computations. `crawlWrap` is a wrapper function for fitting the
343 continuous-time correlated random walk (CTCRW) model of Johnson *et al.* (2008)
344 to one or more tracks (subject to location measurement error and/or temporal ir-
345 regularity) and then predicting temporally-regular tracks of the user's choosing (e.g.

346 15 min, hourly, daily) based on the CTCRW model output. `crawlWrap` returns a
 347 `crwData` object that can be used to draw m realization of the position process within
 348 the `MifitHMM` function. `MifitHMM` is essentially a wrapper function for `fitHMM` that
 349 repeatedly fits the same user-specified HMM to m imputed data sets and stores the
 350 output from each of the m model fits. If a `crwData` object is provided, then `MifitHMM`
 351 will first draw m imputations based on the `crwData` output and then fit the speci-
 352 fied HMM to each imputed data set. If users wish to use a movement model other
 353 than the CTCRW to account for measurement error and temporal irregularity (e.g.
 354 Calabrese *et al.* 2016; Gurarie *et al.* 2017), or if other observation error processes
 355 (e.g. missing data) are to be accounted for in the imputation step, `MifitHMM` can
 356 also be used for analysis of a list of m `momentuHMMData` objects that were imputed
 357 by the user. Based on the m model fits, the `Mipool` function calculates pooled es-
 358 timates, standard errors, and confidence intervals for the working scale parameters,
 359 natural scale parameters (based on transformations of the pooled working param-
 360 eters and mean or user-specified values for any covariates), state sequences, state
 361 probabilities, and activity budgets (i.e. the proportion of the T times step assigned
 362 to each state) using standard multiple imputation formulae (Rubin & Schenker 1986;
 363 McClintock 2017). `Mipool` can be called separately or within `MifitHMM` (using the
 364 `poolEstimates` argument), and the function returns a `miSum` object containing the
 365 pooled output across all imputatons. See sections 3.2, 3.3, and 3.4 for example HMM
 366 analyses that use multiple imputation to account for location measurement error and
 367 temporally irregularity.

368 2.4 Model visualization and diagnostics

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

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

390 2.5 Simulation

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

402 3 Examples

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

409 3.1 African elephant

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

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

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

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

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

431 been created, we're ready to fit a 2-state HMM to the elephant track that includes
 432 temperature effects on the turning angle concentration parameters and cycling tem-
 433 perature effects (with a 24-hour periodicity) on the step length and state transition
 434 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 = "wrpcauchy"),
+                       Par0 = Par0$Par, beta0 = Par0$beta,
+                       DM = DM, formula = formula,
+                       estAngleMean = list(angle=FALSE))
> plot(elephantFit, plotCI = TRUE, covs = data.frame(hour=12))
```

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

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

449 3.2 Northern fur seal

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

Here we specified a gamma distribution for step length ('step'), wrapped Cauchy distribution for turning angle ('angle'), and Poisson distribution for the number of foraging dives ('dive'). The function `getParDM` was used to organize the starting values for the data stream working parameters (`Par0`) in the correct format based on `DM`, `cons`, and estimates of the natural parameters (`Par`) from McClintock *et al.* (2014). As detailed in Supplementary Material, the `DM` and `cons` arguments were specified to avoid label switching among the `nSims` imputed data model fits and enforce similar state-dependent probability distribution constraints as McClintock *et al.* (2014), e.g., constraining the Poisson rate parameters such that the "foraging" state tends to have higher numbers of foraging dives than the "transit" state ($\lambda_2 > \lambda_3$). To prohibit foraging dives for the "resting" state, we used the `fixPar` argument to effectively fix the Poisson rate parameter to zero on the natural scale (i.e. $\lambda_1 \approx 0$). To help deal with the problem of convergence to local maxima, the `retryFits` argument allows users to specify the number of times to attempt to re-fit each model using random perturbations of the parameter estimates as the starting values for optimization.

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

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

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

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

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

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

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

535 **3.4 Grey seal**

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

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

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

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

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

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

```

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

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

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

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

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

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

598 4 Discussion

599 Here we have introduced version 1.0.0 of the R package `momentuHMM` and demon-
600 strated some of its capabilities for conducting multivariate HMM analyses with ani-
601 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

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

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

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

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

656 Acknowledgments

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

661 References

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

672 (2010) Accuracy of argos locations of pinnipeds at-sea estimated using fastloc gps.
 673 *PloS one*, **5**, e8677.

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

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

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

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

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

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

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

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

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

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

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

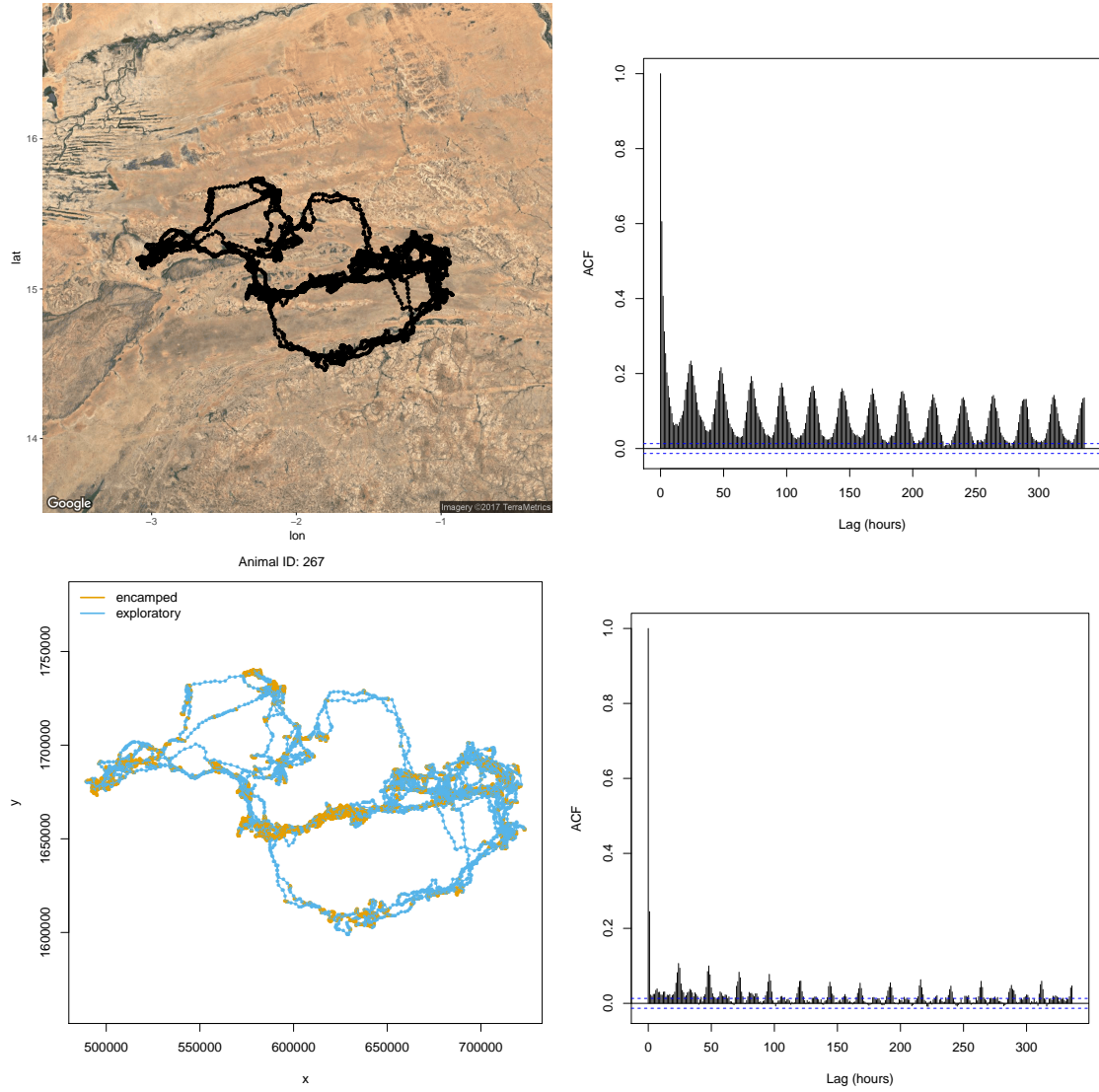


Figure 1. Plot of the elephant track produced using the ‘plotSat’ function (top-left panel), autocorrelation function (ACF) plot of the corresponding step length data (top-right panel), plot of the Viterbi-decoded state sequence for the 2-state (“encamped” and “exploratory”) model generated using the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

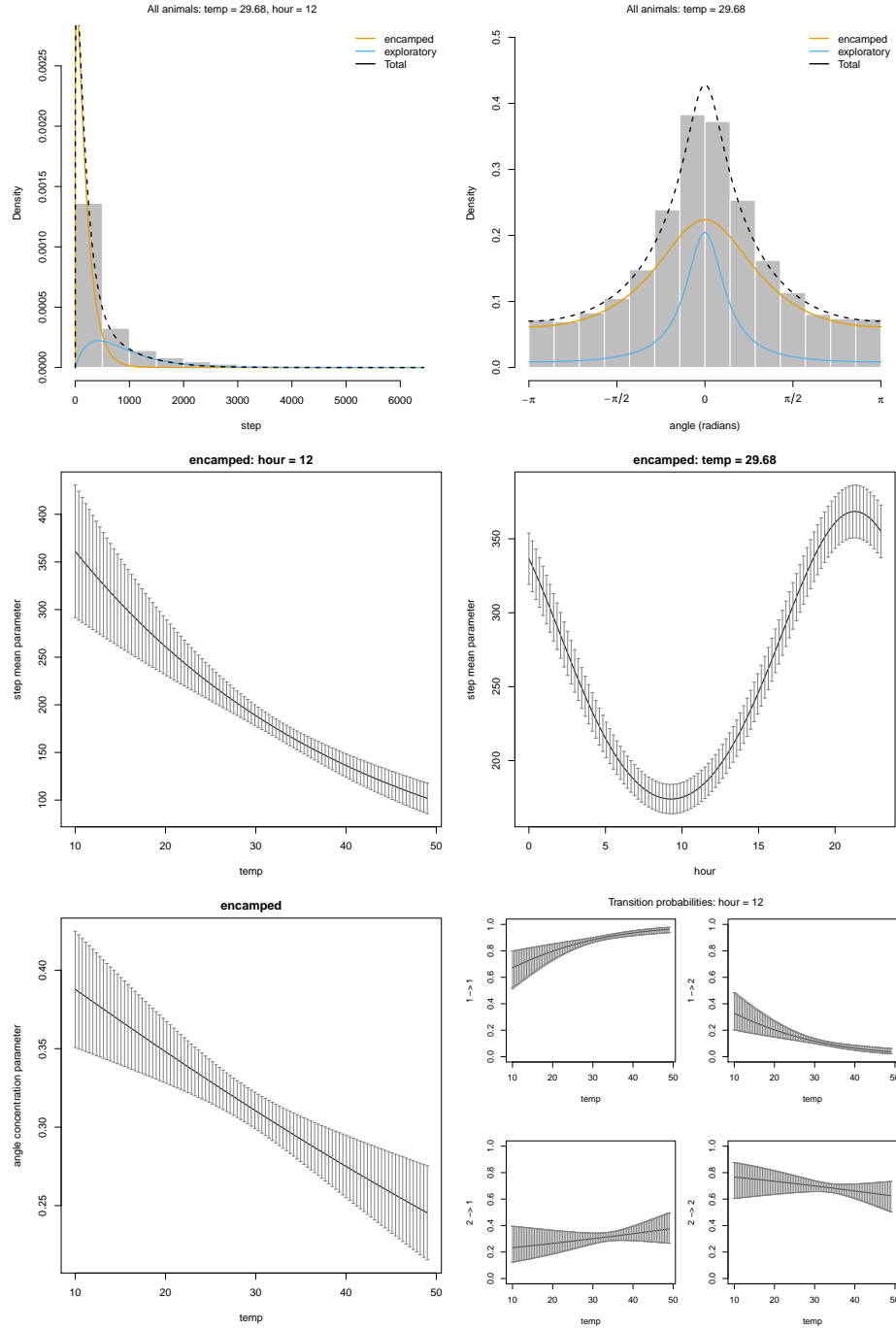


Figure 2. Selected plots for the 2-state (“encamped” and “exploratory”) African elephant example generated using the generic ‘plot’ function. Top panels present histograms of the step length (top-left) and turning angle (top-right) data along with the estimated state-dependent probability distributions 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.

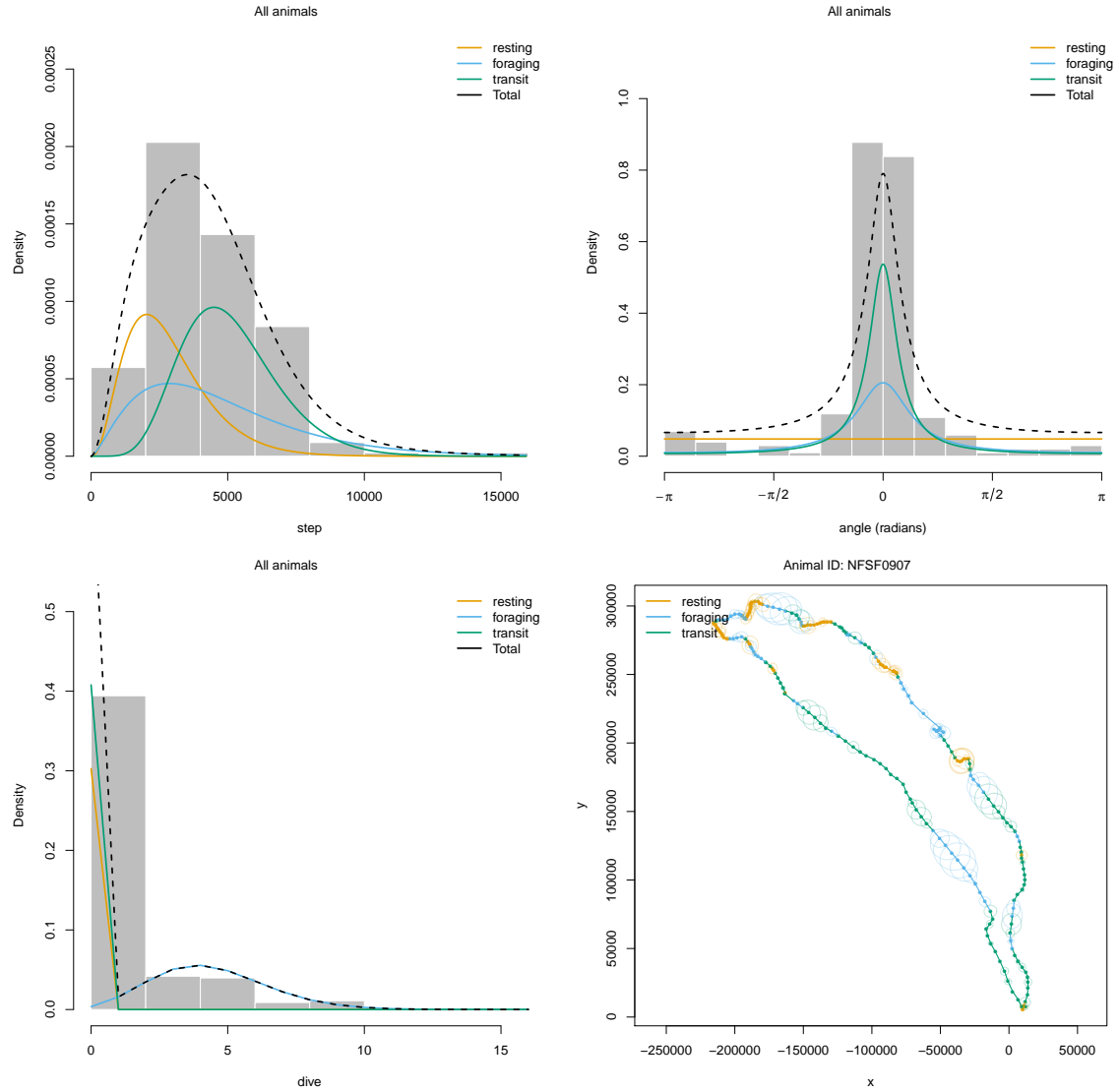


Figure 3. Plots of the northern fur seal example results generated using the generic ‘plot’ function. The estimated probability distributions for step length (top-left panel), turning angle (top-right panel), and number of foraging dives (bottom-left panel) for the 3-state (“resting”, “foraging”, and “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.

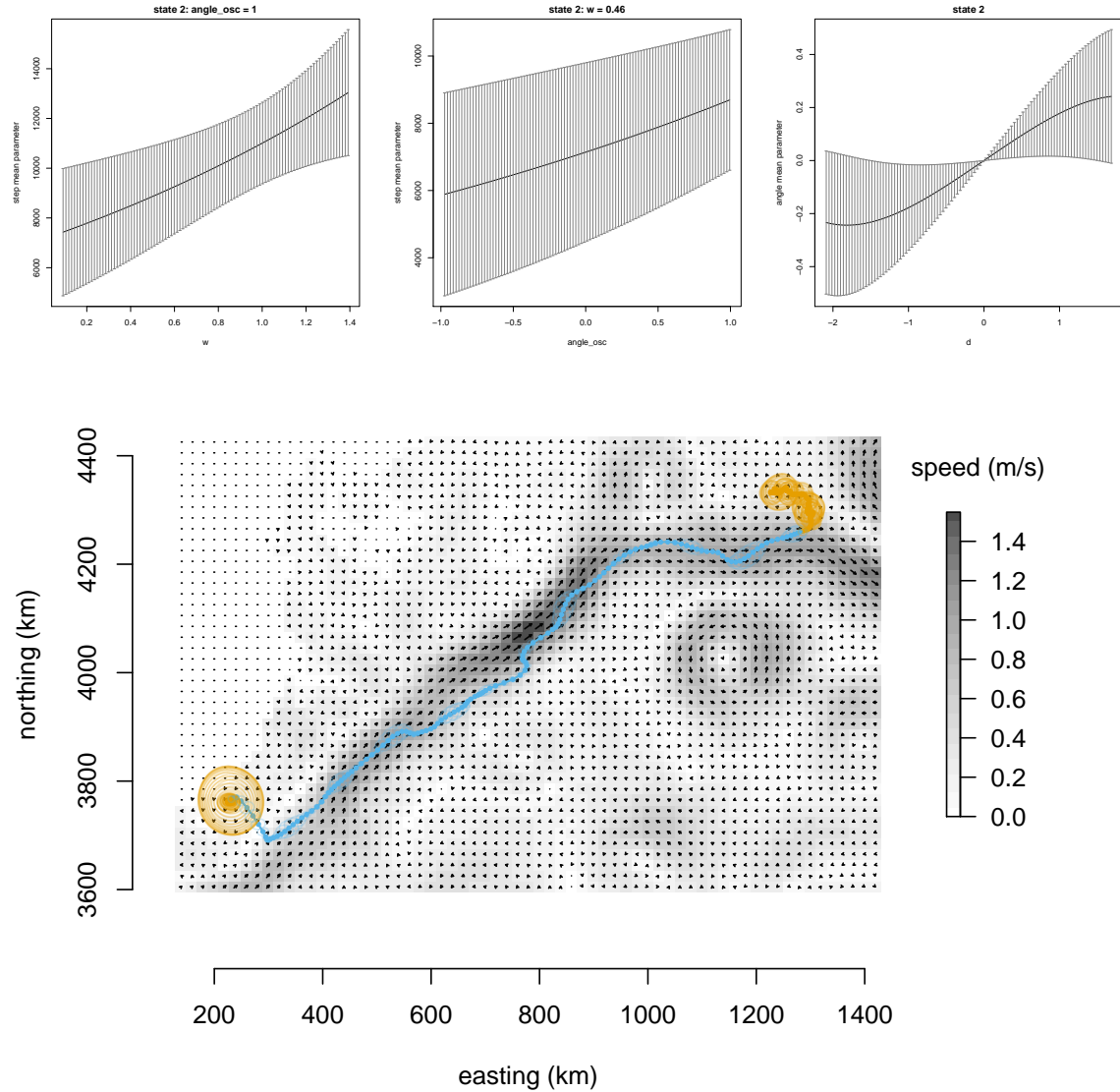


Figure 4. Selected results from the loggerhead turtle example. Top panels include estimates and 95% confidence intervals for the mean step length parameter as a function of ocean surface current speed (w) when ocean surface current direction (r_t) is the same as the bearing (b_t) of movement (i.e. $angle_osc = \cos(b_t - r_t) = 1$; 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.

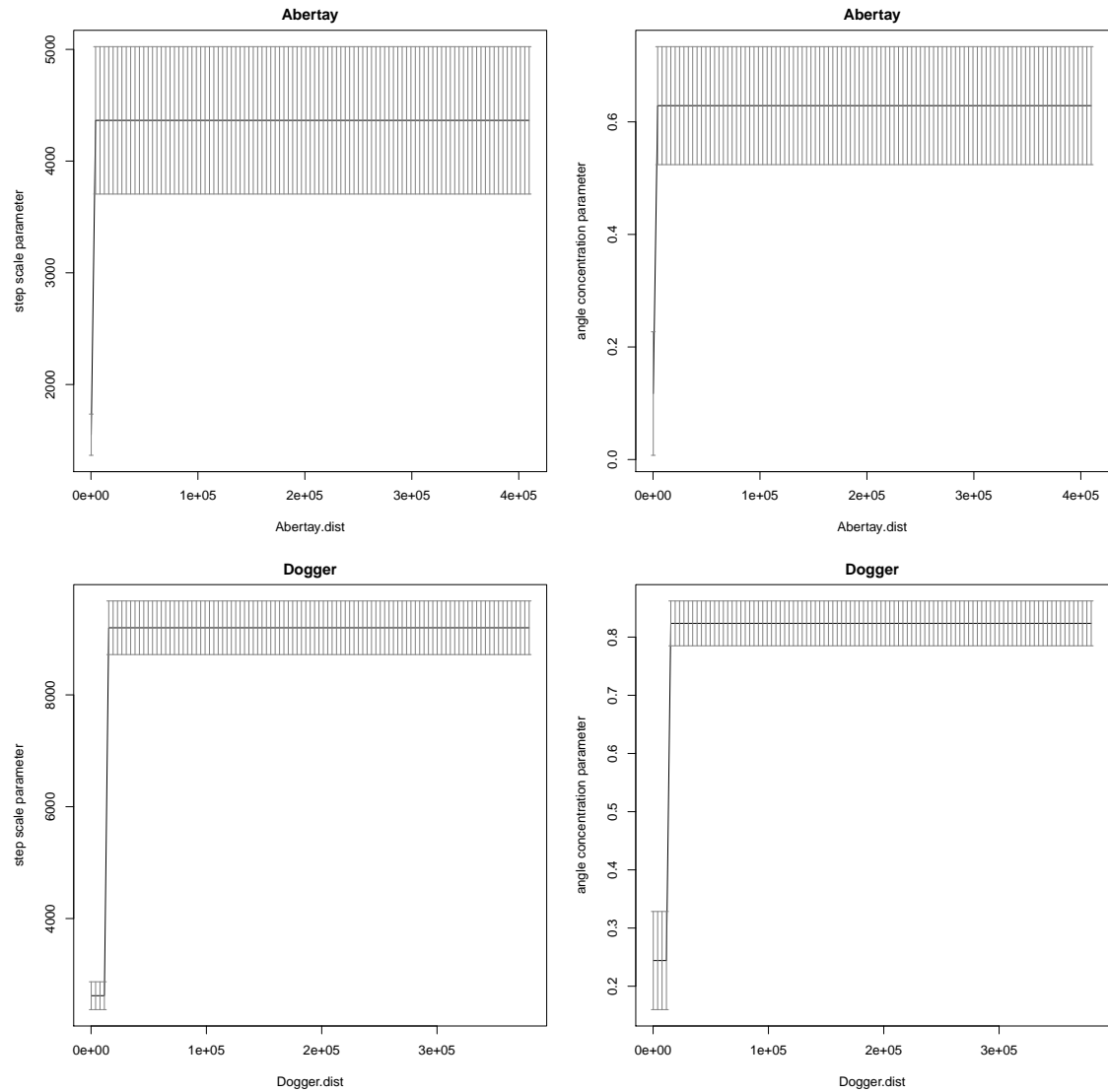


Figure 5. Selected results from the grey seal example. Panels include estimates and 95% confidence intervals for the “Abertay” haul-out state step length scale parameter as a function of distance in meters (‘Abertay.dist’; top-left panel), “Abertay” haul-out state turning angle concentration parameter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

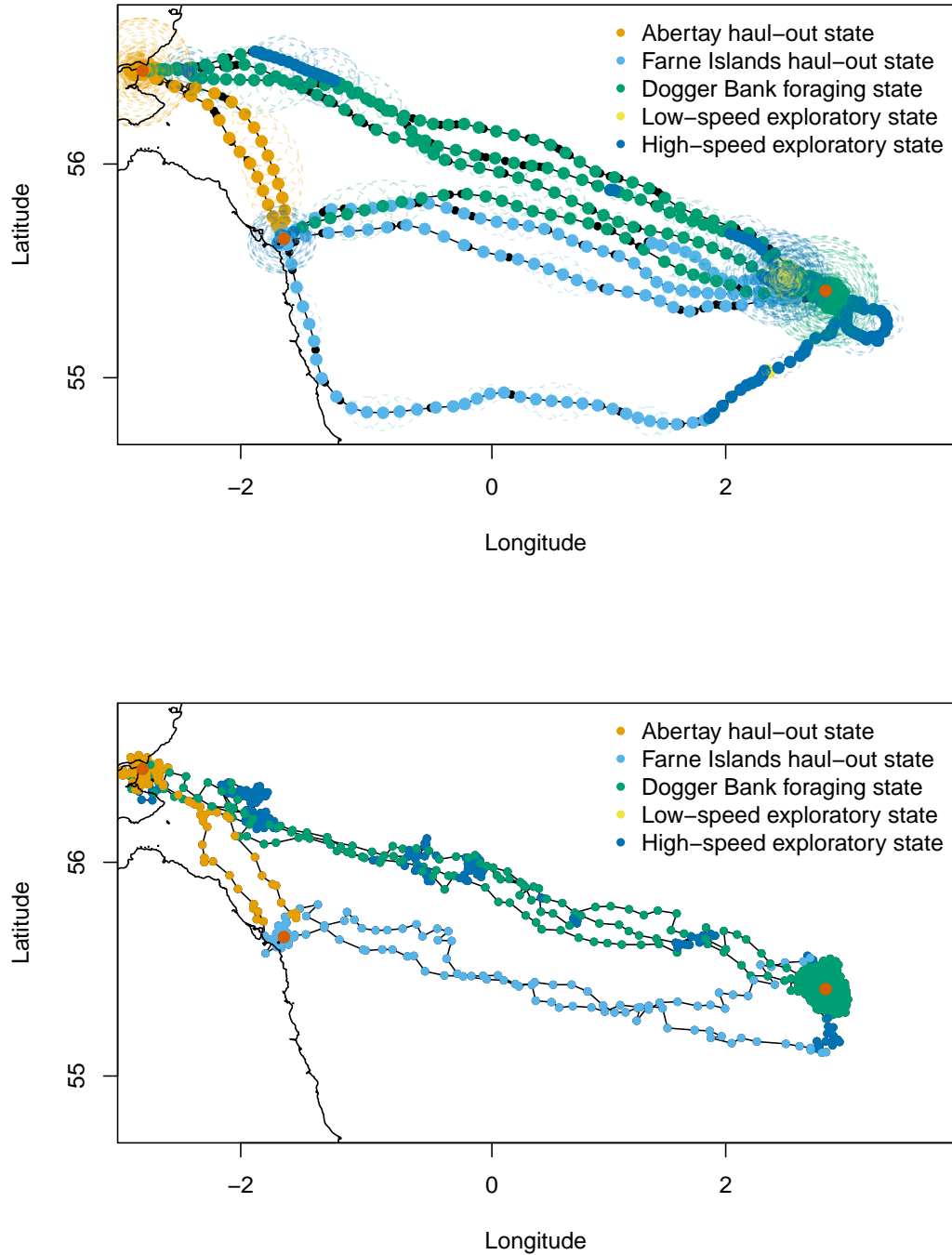


Figure 6. Fitted and simulated tracks from the grey seal example. This seal tended to move in a clockwise fashion between two haul-out locations (“Abertay” and “Farne Islands”) and a foraging area (“Dogger Bank”) in the North Sea. Top panel plots the pooled track, 95% error ellipse confidence bands, 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.