moveHMM and momentuHMM

Analysing animal movement data in R

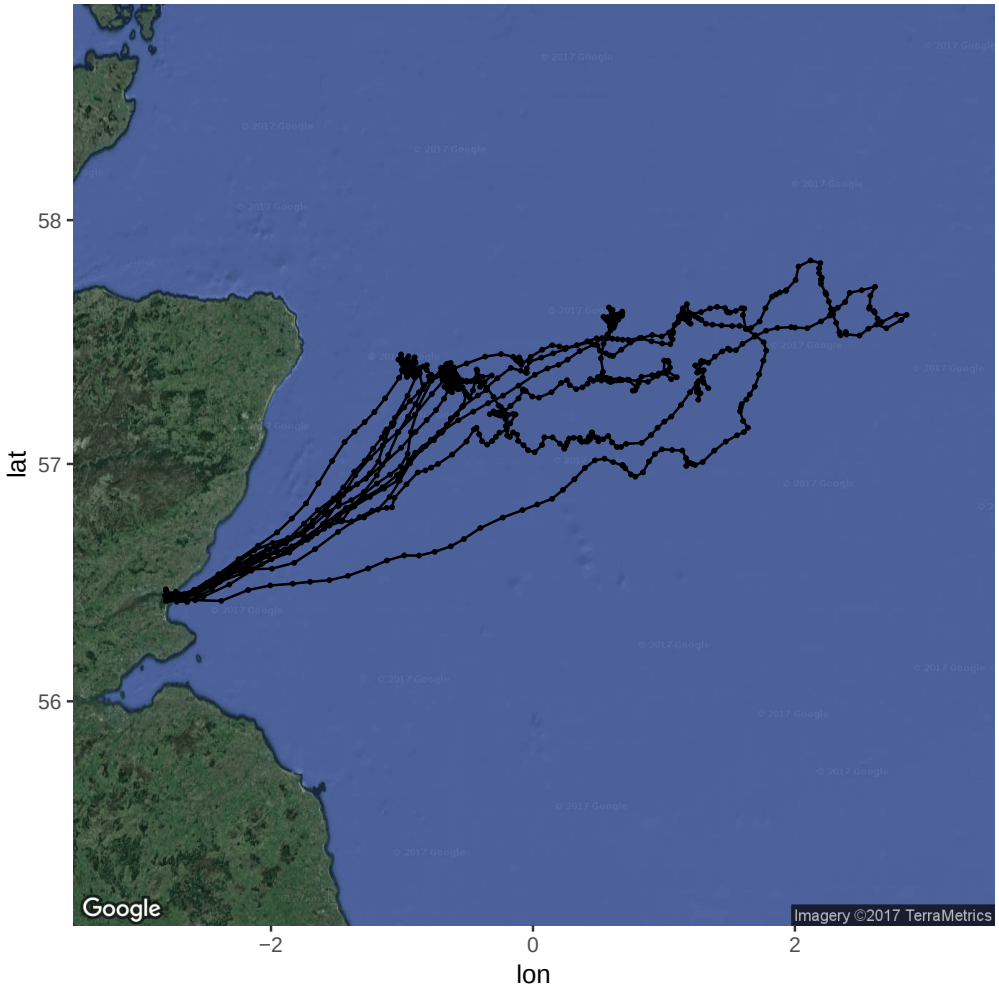
Théo Michelot[1], Roland Langrock[2]

[1] *University of Sheffield,* [2] *Universität von Bielefeld*

18 December 2017

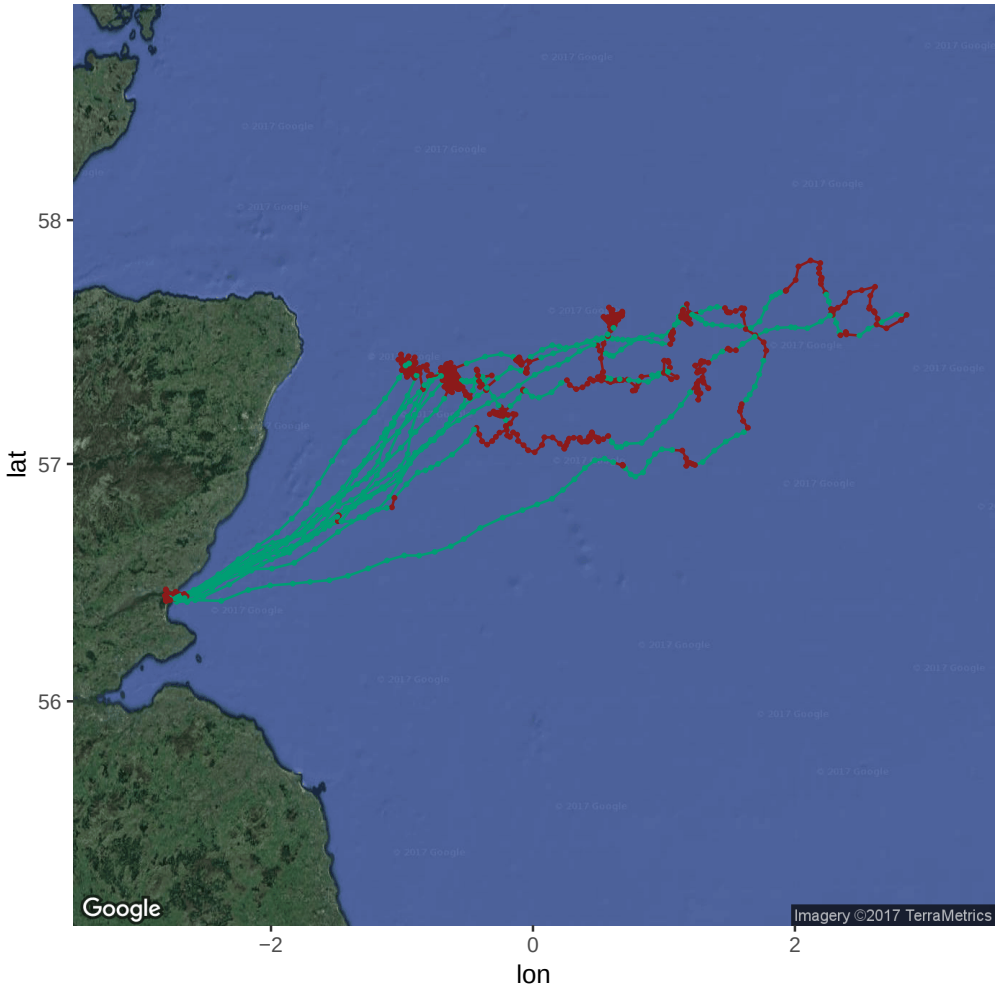
Background on hidden Markov models

HMM of animal movement



Data from: Russell et al. (2015), "Intrinsic and extrinsic drivers of activity budgets in sympatric grey and harbour seals", Oikos, 124(11).

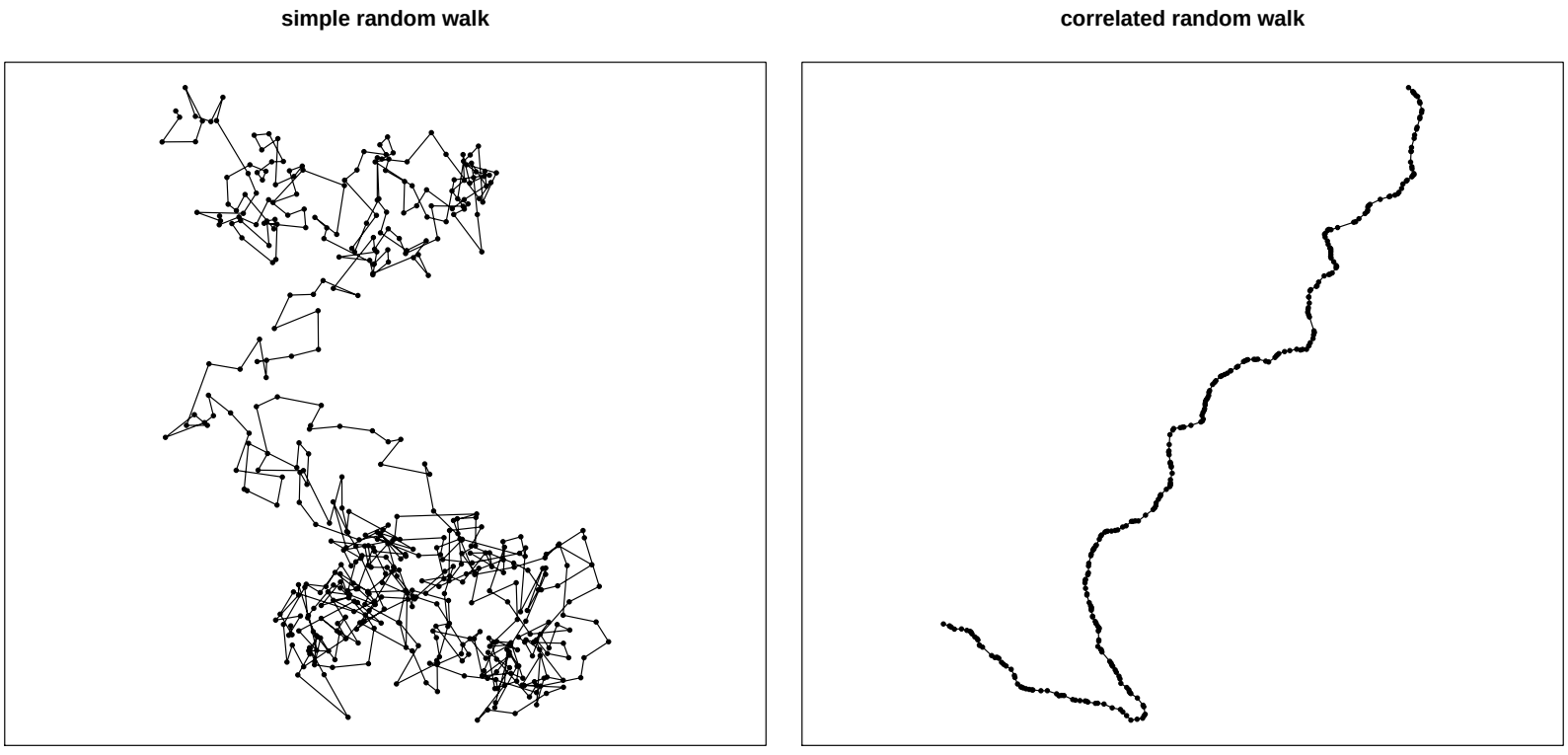
HMM of animal movement



Data from: Russell et al. (2015), "Intrinsic and extrinsic drivers of activity budgets in sympatric grey and harbour seals", Oikos, 124(11).

Correlated random walk

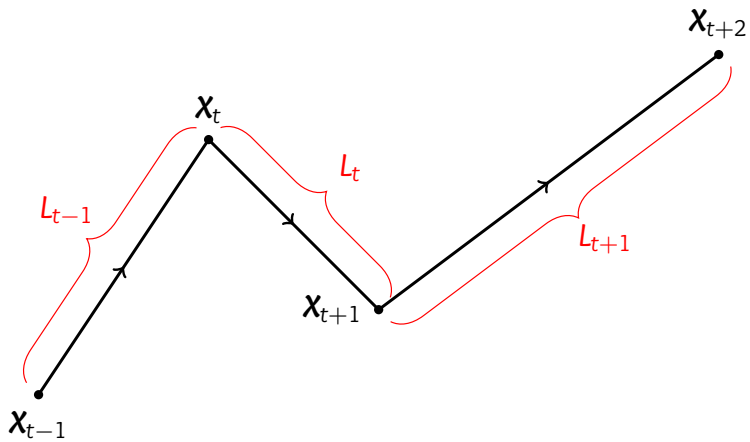
A correlated random walk includes **persistence in direction**.
$\longrightarrow$ Correlation between successive directions.

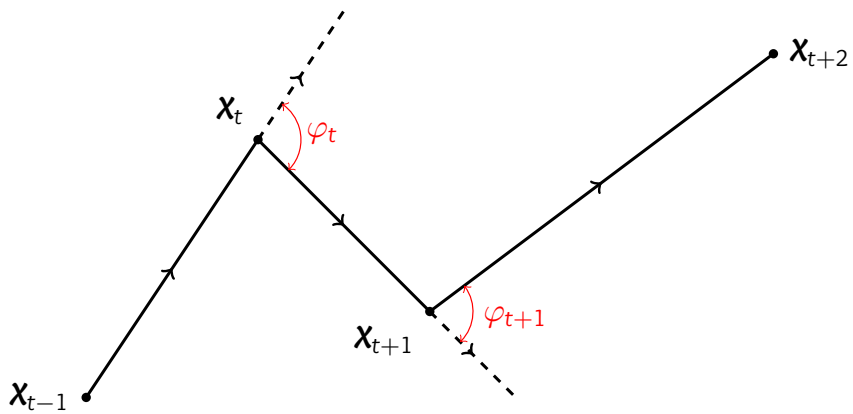


Movement metrics

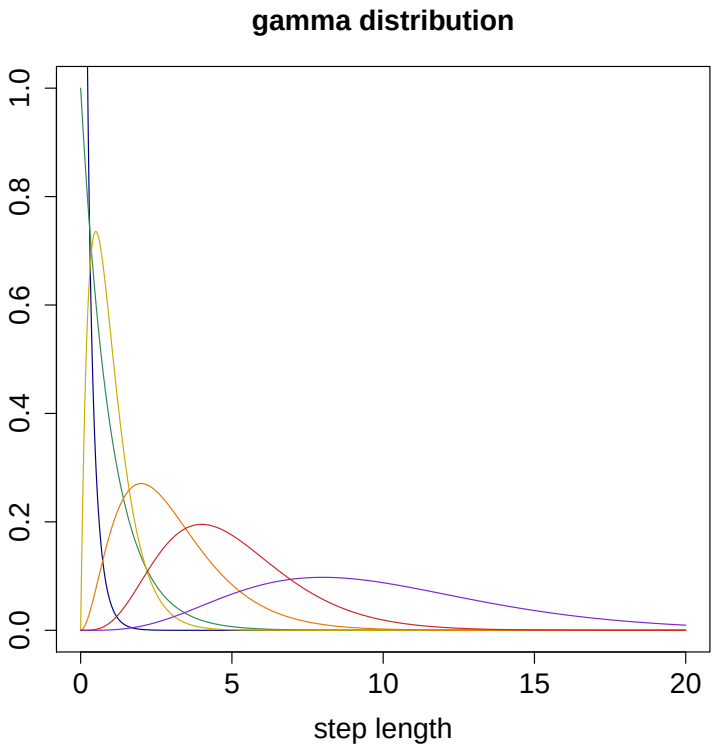
In a correlated random walk, we can model:

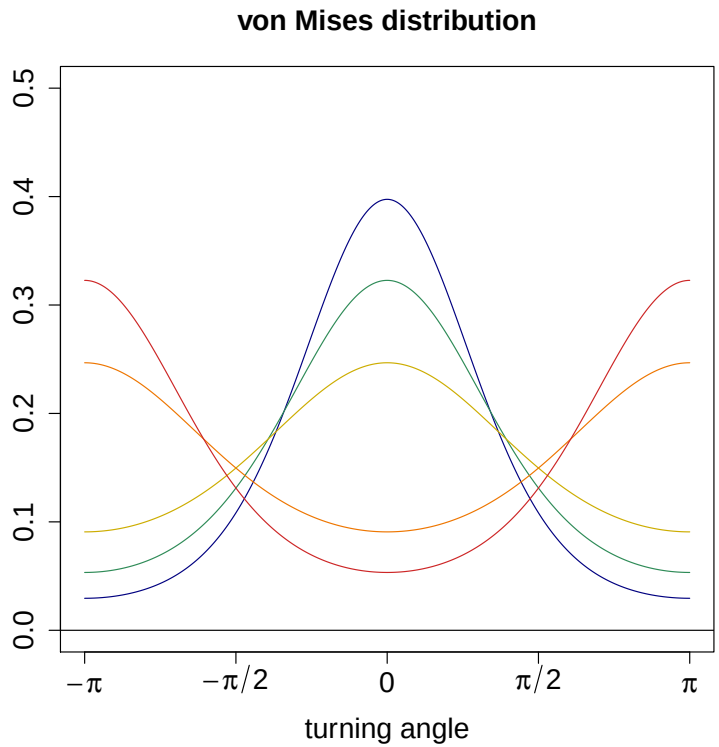
- step lengths (L_t);
- turning angles (φ_t).





Modelling the steps and angles





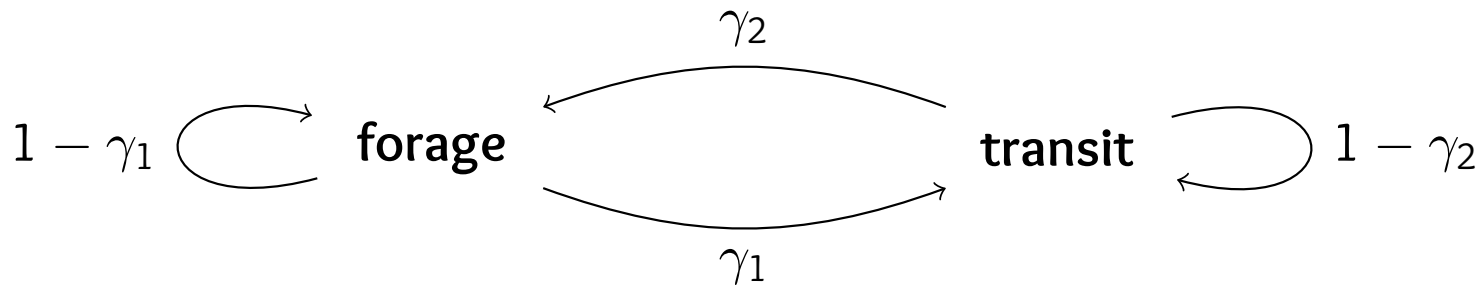
Multistate random walk

Idea: the animal switches between several movement processes, corresponding to several **behaviours**.
$\longrightarrow$ Behavioural process = unobserved Markov chain (S_t).

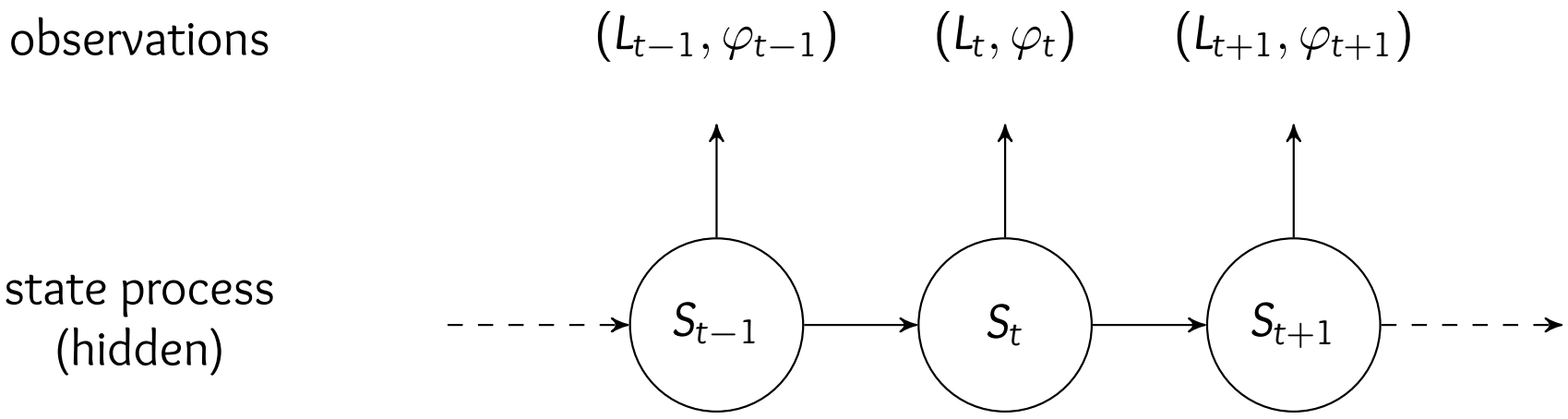
Multistate random walk

Idea: the animal switches between several movement processes, corresponding to several **behaviours**.
$\longrightarrow$ Behavioural process = unobserved Markov chain (S_t).

Example:



Hidden Markov model for animal movement



The steps and angles are modelled by state-dependent distributions.
For example:

$$L_t | S_t = j \sim \text{gamma}(\alpha_j, \beta_j)$$
$$\varphi_t | S_t = j \sim \text{von Mises}(\mu_j, \kappa_j)$$

Covariates

Does [*insert covariate*] have an effect on the probability that the animal is [*insert behaviour*]?

$\longrightarrow$ Time-varying transition probabilities.

In a 2-state model:

$$\begin{cases} \Pr(S_{t+1} = 2 | S_t = 1) = \text{logit}^{-1}(\beta_0^{(12)} + \sum_{i=1}^{m} \beta_i^{(12)} w_{i,t}) \\ \Pr(S_{t+1} = 1 | S_t = 2) = \text{logit}^{-1}(\beta_0^{(21)} + \sum_{i=1}^{m} \beta_i^{(21)} w_{i,t}) \end{cases}$$

moveHMM

Introduction to moveHMM

The package is available on CRAN:

```
install.packages("moveHMM")
```

A good place to start is the **package vignette** (background, detailed case study with code, implementation details...), and the **package documentation** (details for each function).

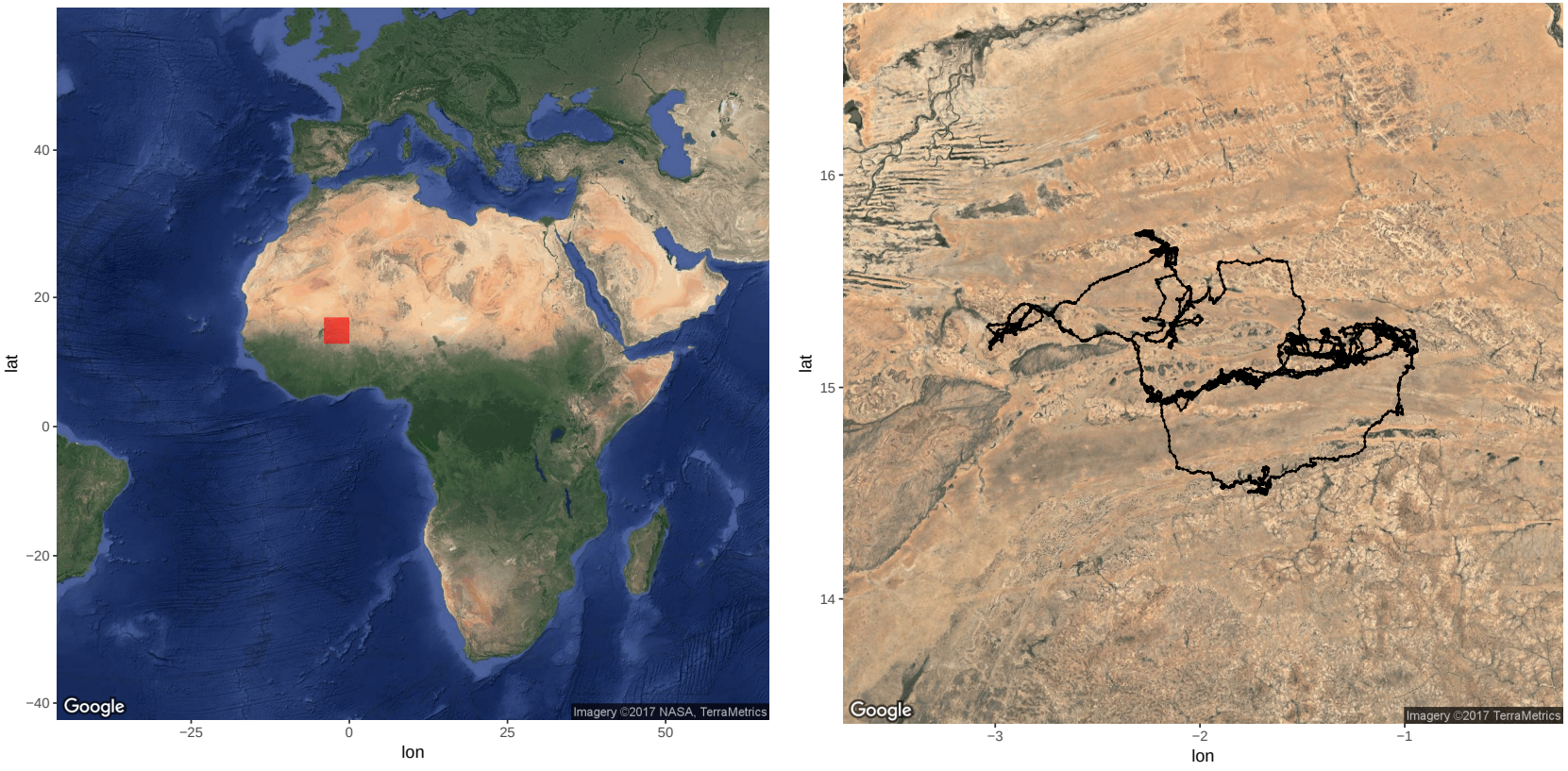
Michelot, Langrock, Patterson (2016). moveHMM: An R package for the statistical modelling of animal movement data using hidden Markov models, *MEE*.

moveHMM/momentuHMM workflow

1. Visualise the data.
2. Choose model formulation:
   $\longrightarrow$ how many states?
   $\longrightarrow$ which distributions for the steps/angles?
   $\longrightarrow$ any covariates?
3. Fit model(s). (Fast!)
4. Visualise model:
   $\longrightarrow$ map of "decoded" tracks;
   $\longrightarrow$ covariate effects.
5. Visualise diagnostics (pseudo-residuals).

Elephant case study

Hourly locations over one year + temperature recordings.



Wall et al. (2014), "Elliptical time-density model to estimate wildlife utilization distributions" Methods in Ecology and Evolution, 5 (780–790).

(From the Movebank data repository.)

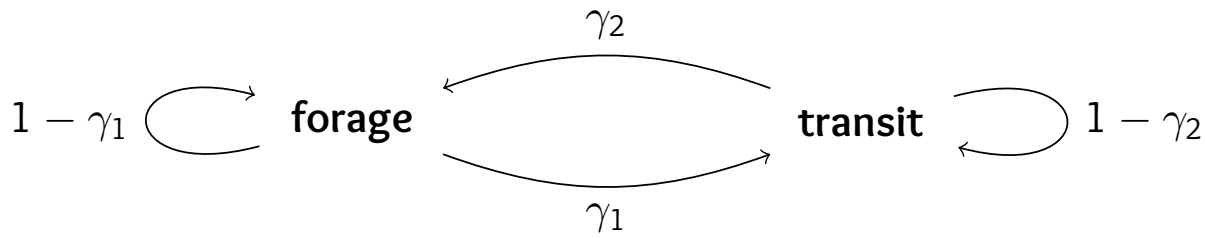
1. Prepare the data
2. Fit the model
3. Visualise the results
4. Include covariates

Formatting the data

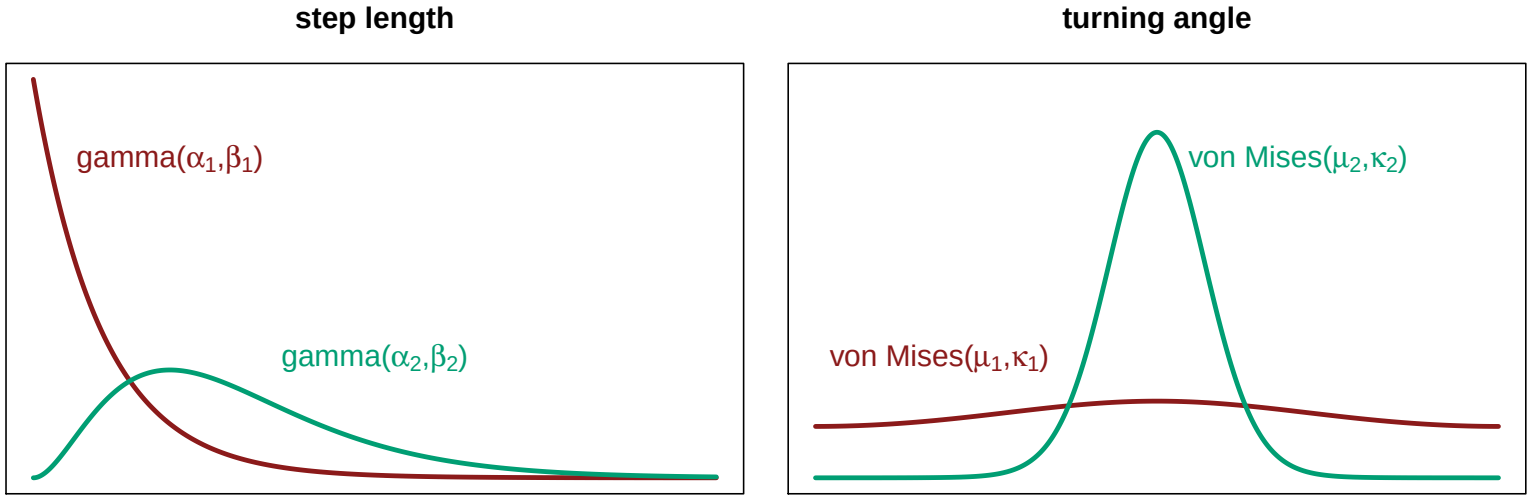
```
track <- read.csv("elephant.csv")

head(track)

##   ID          x         y temp tod
## 1  1 -2.160167 15.65350   38  17
## 2  1 -2.160075 15.65452   35  18
## 3  1 -2.159902 15.65451   32  19
## 4  1 -2.159435 15.65489   30  20
## 5  1 -2.158113 15.65512   29  21
## 6  1 -2.157848 15.65461   28  22
```

Regular sampling frequency

- Transition probabilities assume regular time intervals.



- State-dependent distributions assume regular time intervals.



$\longrightarrow$ We need **regular time intervals** between data rows.

prepData

```
library(moveHMM)

data <- prepData(track)

## Warning in prepData(track):  There are 15 missing covariate values.
Each will be replaced by the closest available value.

head(data)

##   ID       step      angle         x        y temp tod
## 1  1 0.11329503         NA -2.160167 15.65350   38  17
## 2  1 0.01862565 -1.5534731 -2.160075 15.65452   35  18
## 3  1 0.06477228  0.7575722 -2.159902 15.65451   32  19
## 4  1 0.14400818 -0.5090123 -2.159435 15.65489   30  20
## 5  1 0.06253091 -1.2781269 -2.158113 15.65512   29  21
## 6  1 0.20910448  2.8261263 -2.157848 15.65461   28  22
```

prepData

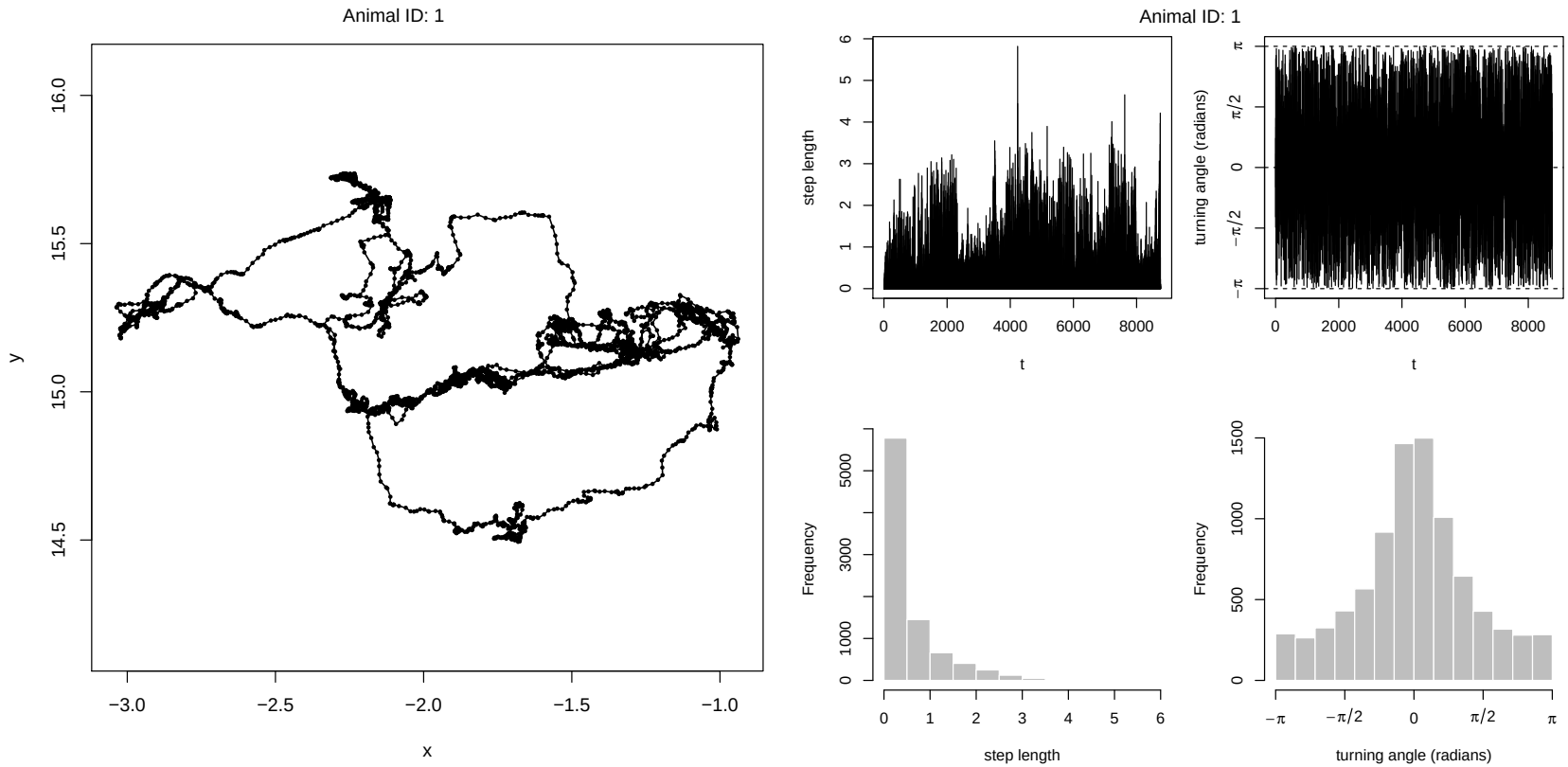
```r
# longitude-latitude (default)
data <- prepData(track, type="LL")
```

$\longrightarrow$ Step lengths computed with `spDistsN1` (package sp).
$\longrightarrow$ Turning angles computed with `bearing` (package geosphere).

```r
# Easting-Northing
data <- prepData(track, type="UTM")
```

Visualise the data

```
plot(data, ask=FALSE)
```



1. Prepare the data
2. Fit the model
3. Visualise the results
4. Include covariates

fitHMM

```r
# define initial parameters
stepMean0 <- c(0.1,1) # mean of step length distribution (one for each state)
stepSD0 <- c(0.2,0.2) # SD of step length distribution
angleMean0 <- c(0,0) # mean of angle distribution
angleCon0 <- c(0.5,5) # concentration of angle distribution

# fit 2-state model
m <- fitHMM(data, nbStates=2, stepPar0=c(stepMean0,stepSD0),
            anglePar0=c(angleMean0,angleCon0), verbose=2)
```

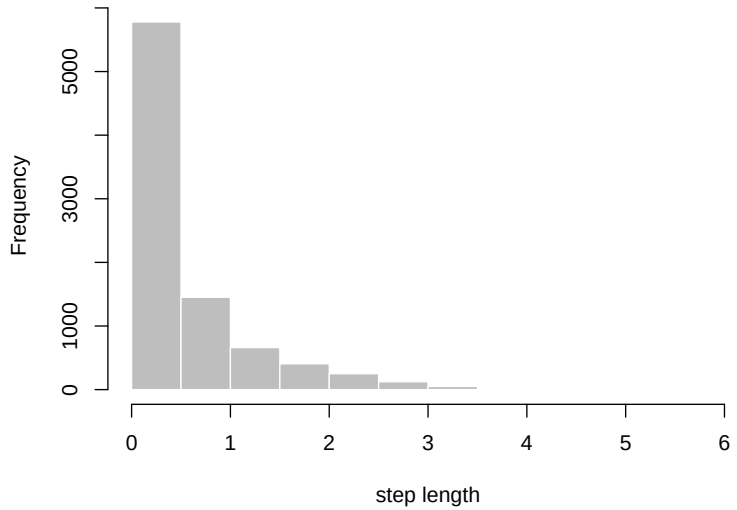
Fitted model

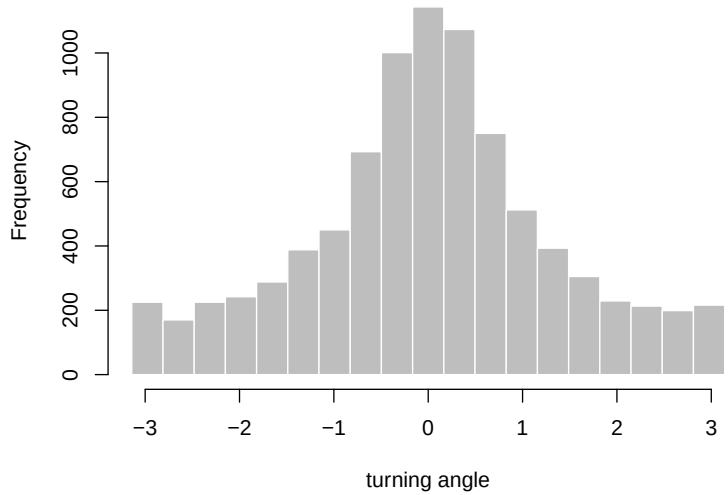
```
m
```

```
## Value of the maximum log-likelihood: -15412.39
##
## Step length parameters:
## ----------------------
##        state 1  state 2
## mean 0.2247176 1.208059
## sd   0.2400380 0.774237
##
## Turning angle parameters:
## ------------------------
##                   state 1     state 2
## mean          0.07109104 0.002108457
## concentration 0.56670205 2.465596954
##
## Regression coeffs for the transition probabilities:
## --------------------------------------------------
##              1 -> 2    2 -> 1
## intercept -2.244511 -1.299499
##
## Transition probability matrix:
## -----------------------------
##           [,1]       [,2]
## [1,] 0.9041760 0.09582395
## [2,] 0.2142493 0.78575067
##
## Initial distribution:
## --------------------
## [1] 9.999935e-01 6.457625e-06
```

fitHMM: initial parameters

1. Plot histograms of step lengths and turning angles.





2. "What are some plausible values for the parameters?"

```
stepMean0 <- c(0.1,1) # mean of step length distribution
stepSD0 <- c(0.2,0.2) # SD of step length distribution
angleMean0 <- c(0,0) # mean of angle distribution
angleCon0 <- c(0.5,5) # concentration of angle distribution
```

3. Try many different initial parameters, maybe chosen at random.

fitHMM: number of states

There is no general method to select the "optimal" number of states.

1. Fit 2-state model, 3-state model, etc., and compare them:
   - Model comparison with AIC/BIC/...

```
AIC(mod2,mod3,mod4)
```

   - Model checking using pseudo-residuals.

2. **Biological interpretation!**

Pohle et al. (2017). Selecting the number of states in hidden Markov models: pragmatic solutions illustrated using animal movement, *JABES*.

fitHMM: other modelling choices

- Choice of distributions for the steps and angles.

```
m <- fitHMM(data, nbStates=2,
            stepDist="weibull", angleDist="wrpcauchy",
            stepPar0=c(stepShape0,stepScale0),
            anglePar0=c(angleMean0,angleCon0))
```

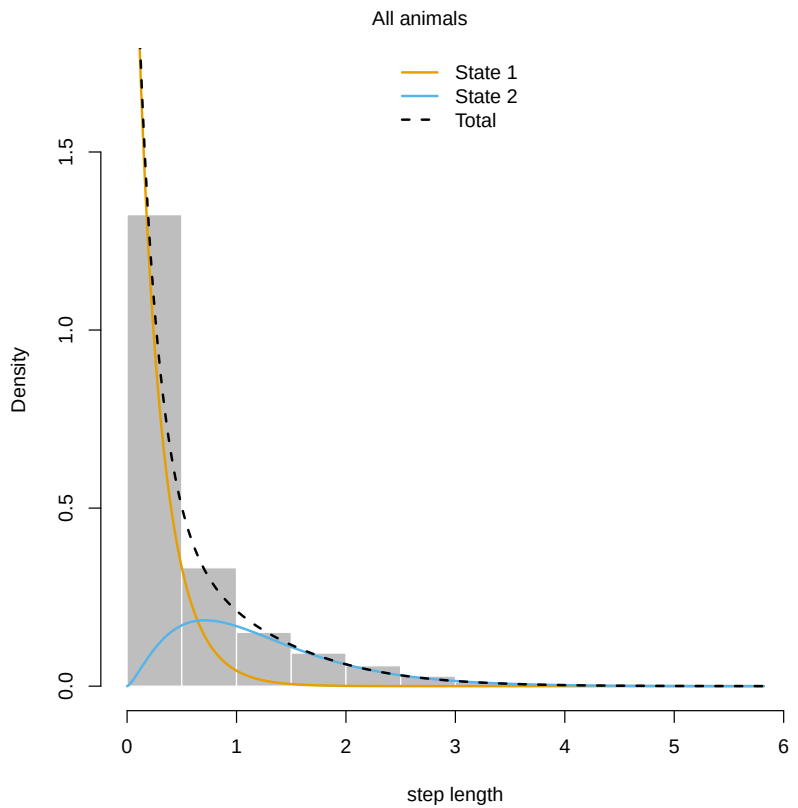
- Covariates on the switching probabilities.

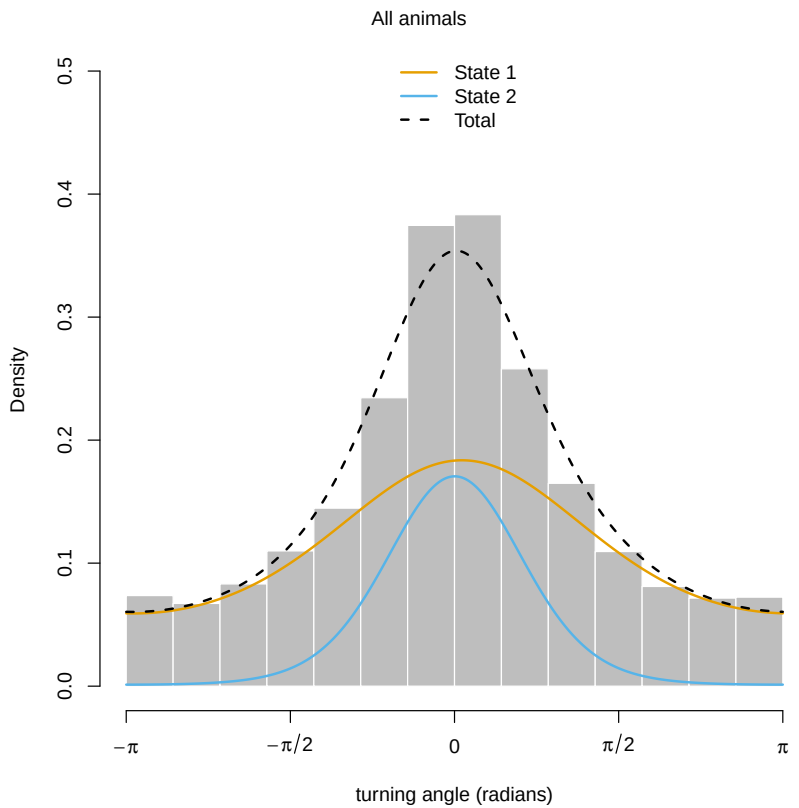
```
m <- fitHMM(data, nbStates=2, stepPar0=c(stepMean0,stepSD0),
            anglePar0=c(angleMean0,angleCon0),
            formula=~cov1+cov2)
```

1. Prepare the data
2. Fit the model
3. Visualise the results
4. Include covariates

Plot a fitted model

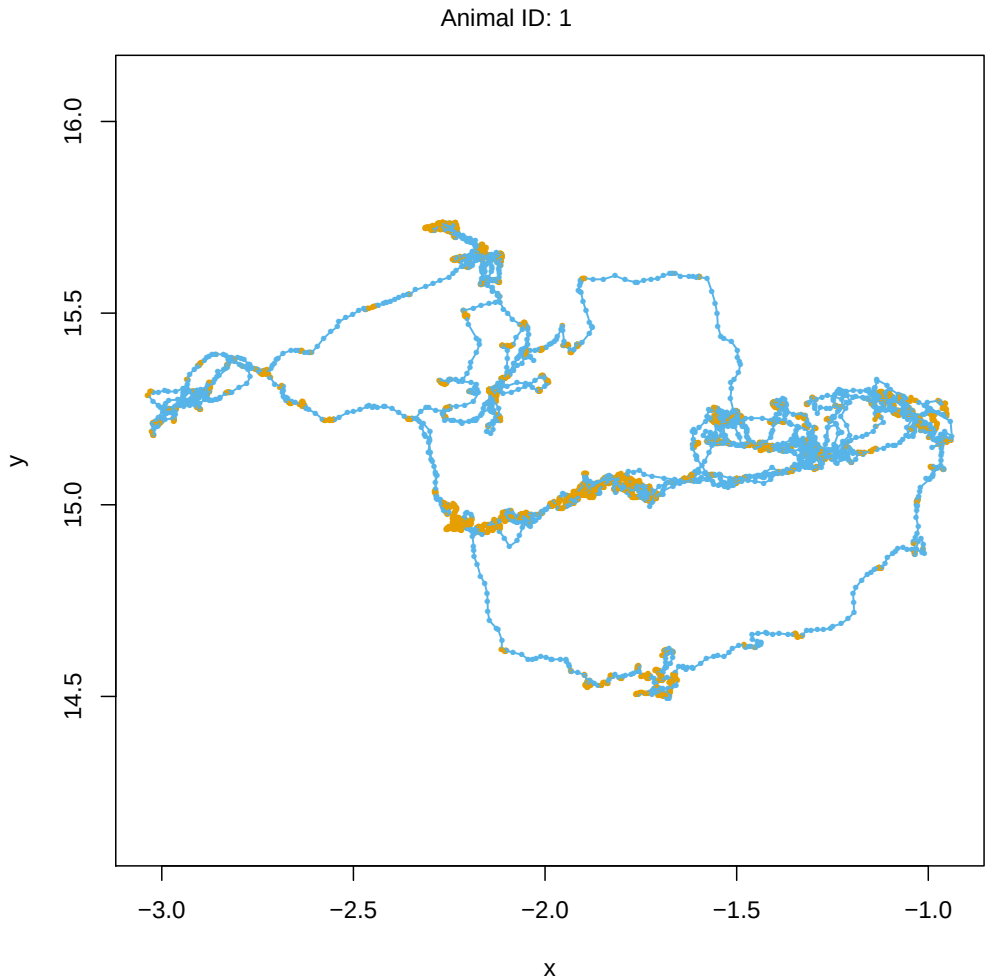
```
plot(m)
```





Plot a fitted model

```
plot(m)
```

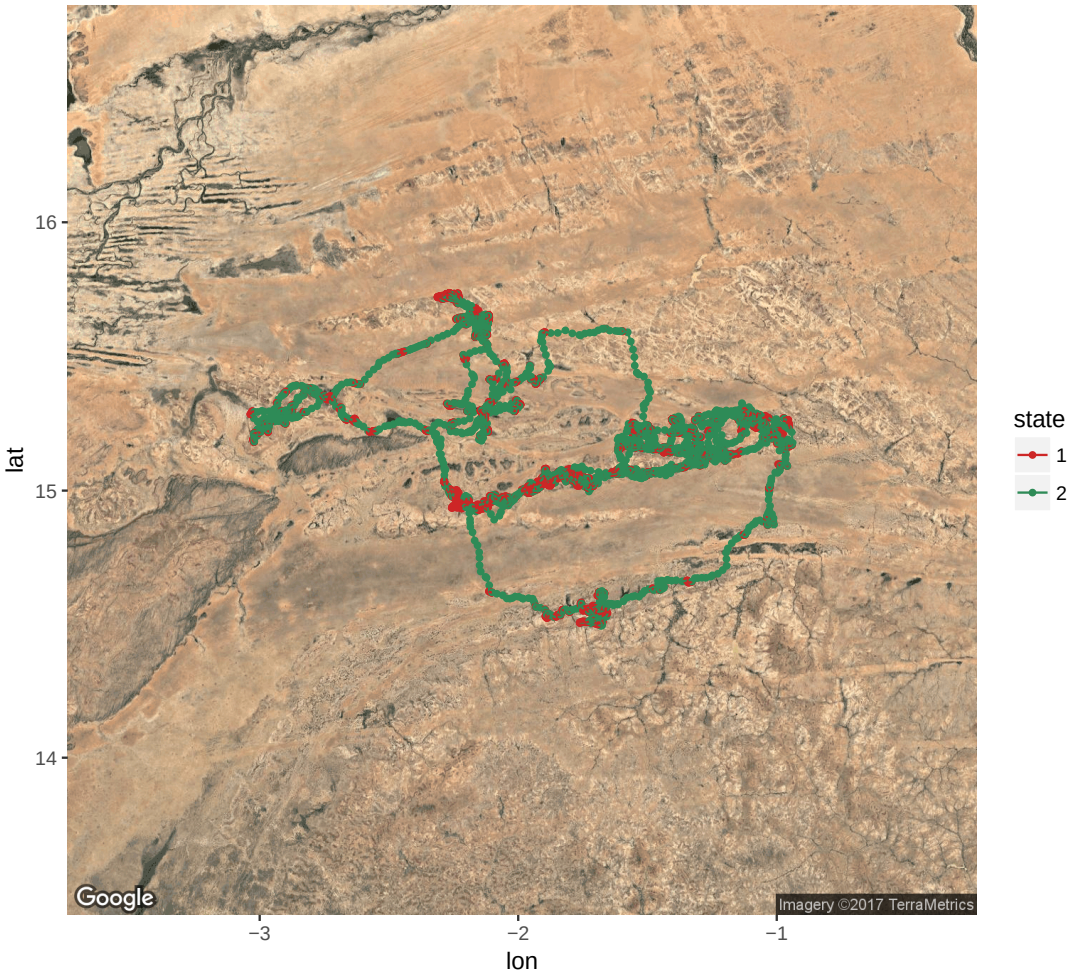


Decode the state process

```
states <- viterbi(m)

head(states)

## [1] 1 1 1 1 1 1

sp <- stateProbs(m)

head(sp)

##              [,1]         [,2]
## [1,] 0.9999999 9.115038e-08
## [2,] 0.9999721 2.788889e-05
## [3,] 0.9991467 8.532958e-04
## [4,] 0.9957130 4.286970e-03
## [5,] 0.9996256 3.744054e-04
## [6,] 0.9997358 2.642377e-04
```

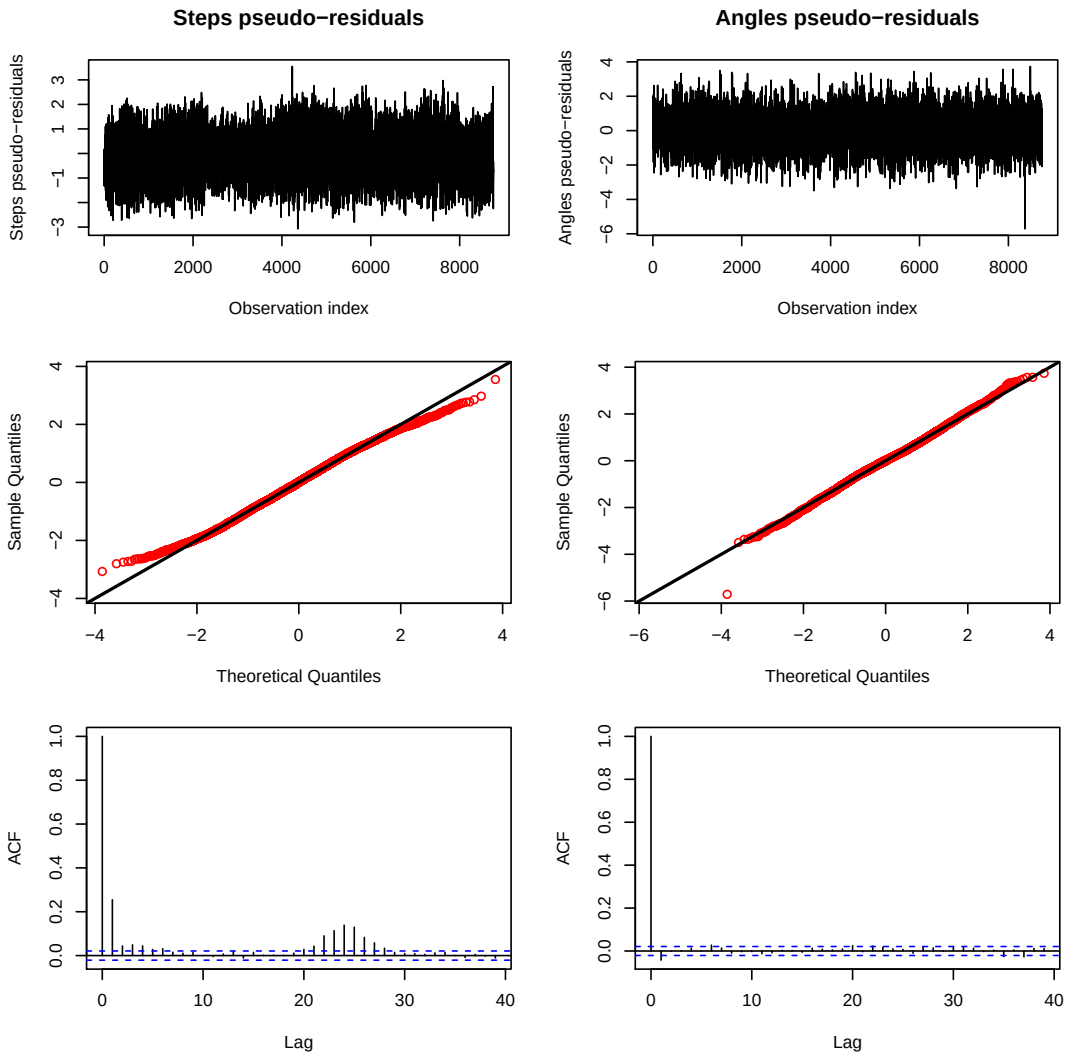
Plot the decoded track

```
plotSat(data, zoom=8, states=states,
        col = c("firebrick3","seagreen4"), ask=FALSE)
```

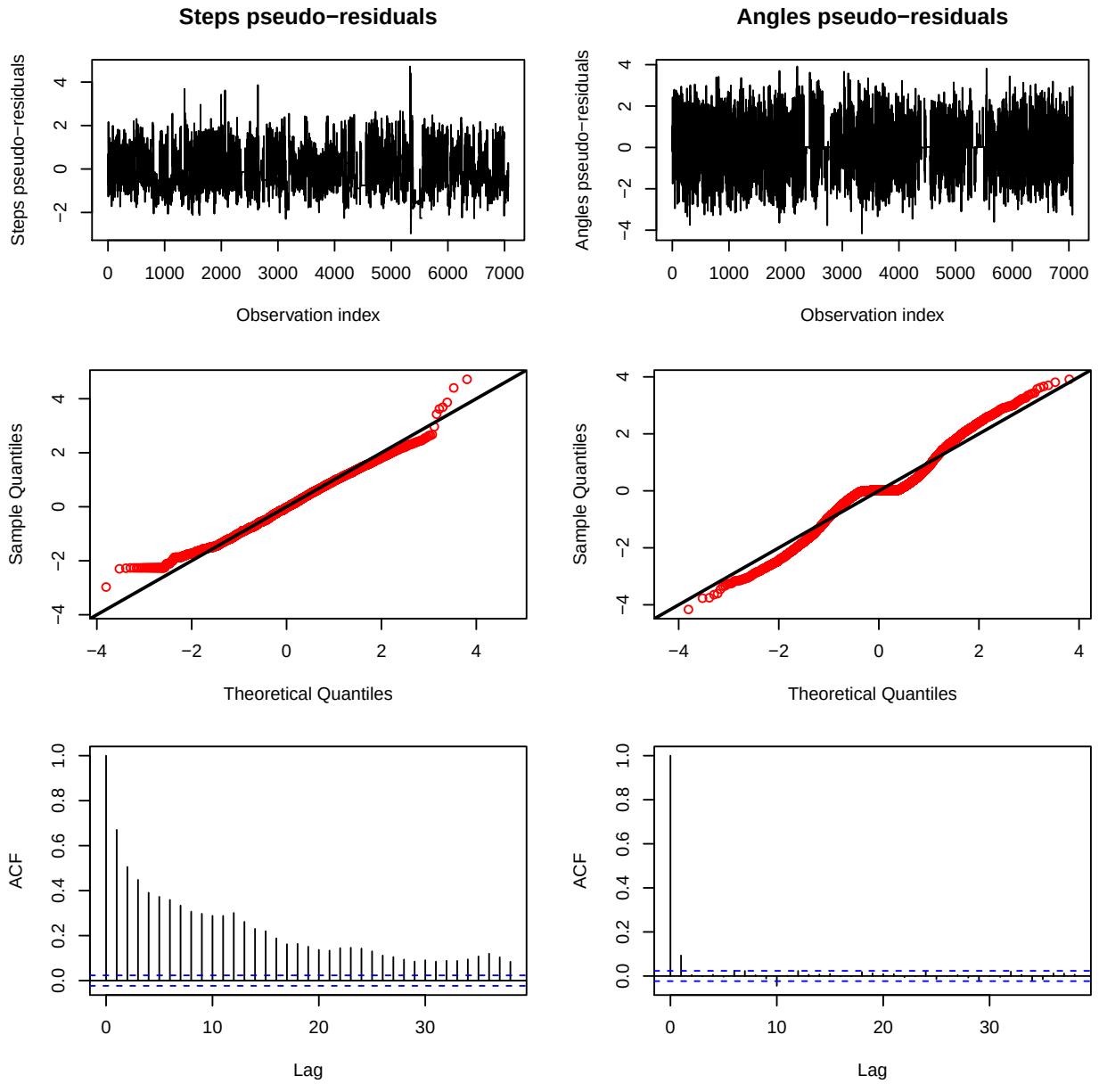


Pseudo-residuals

```
plotPR(m)
```

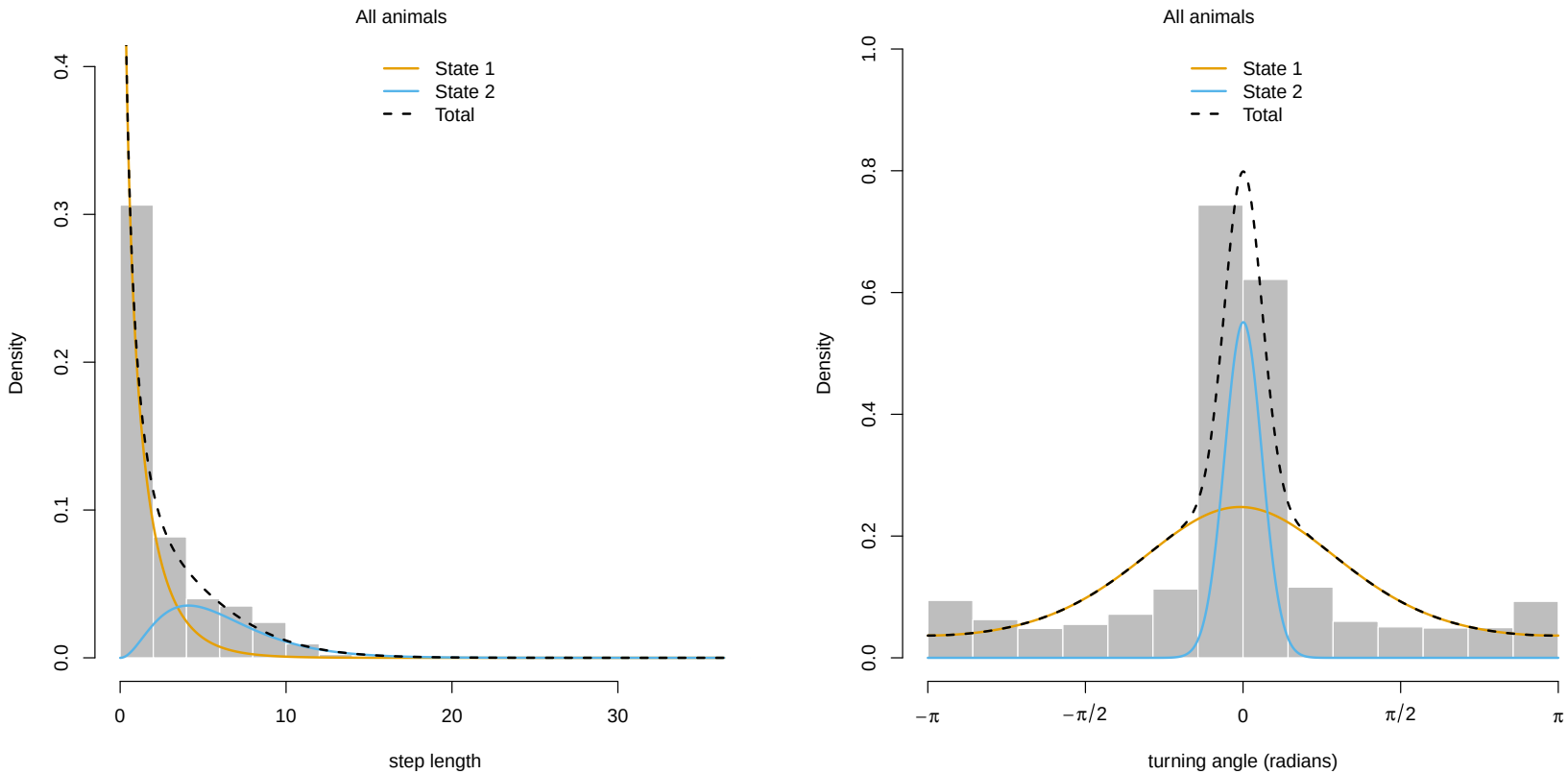


Pseudo-residuals: grey seal



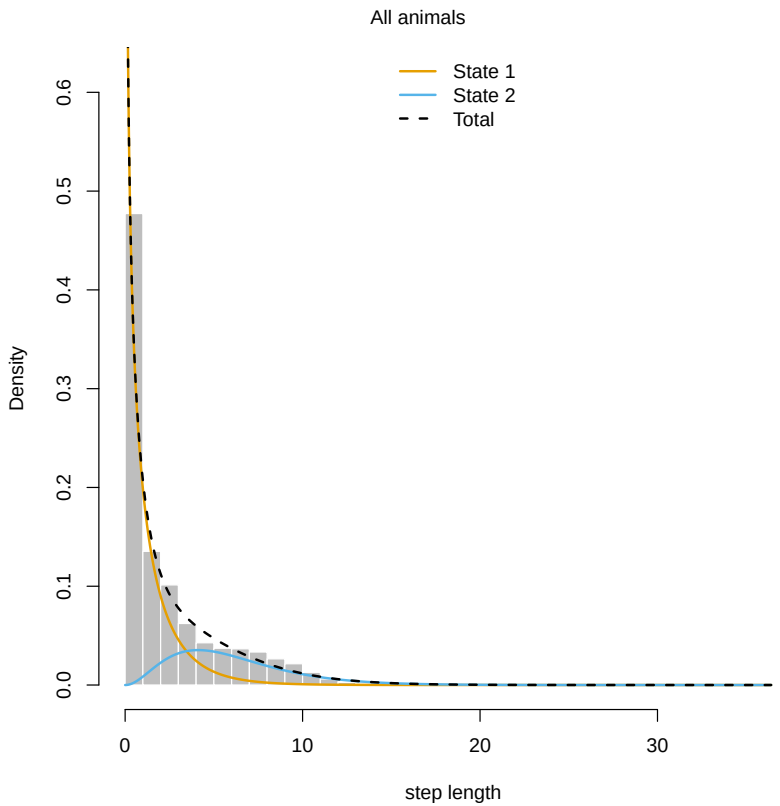
Pseudo-residuals: grey seal

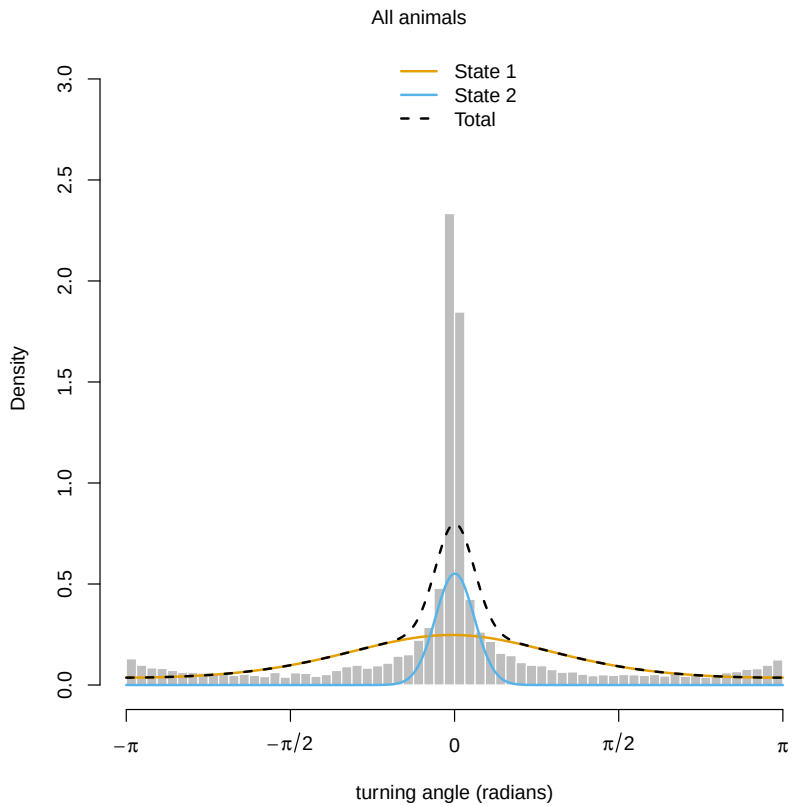
```
plot(seal_model)
```



Pseudo-residuals: grey seal

```
plot(seal_model, breaks=50)
```





1. Prepare the data
2. Fit the model
3. Visualise the results
4. Include covariates

Covariates in transition probabilities

In `fitHMM`, the argument "formula" is a standard R formula, e.g.

- `formula=~cov1`
- `formula=~cov1+cov2`
- `formula=~cov1*cov2`

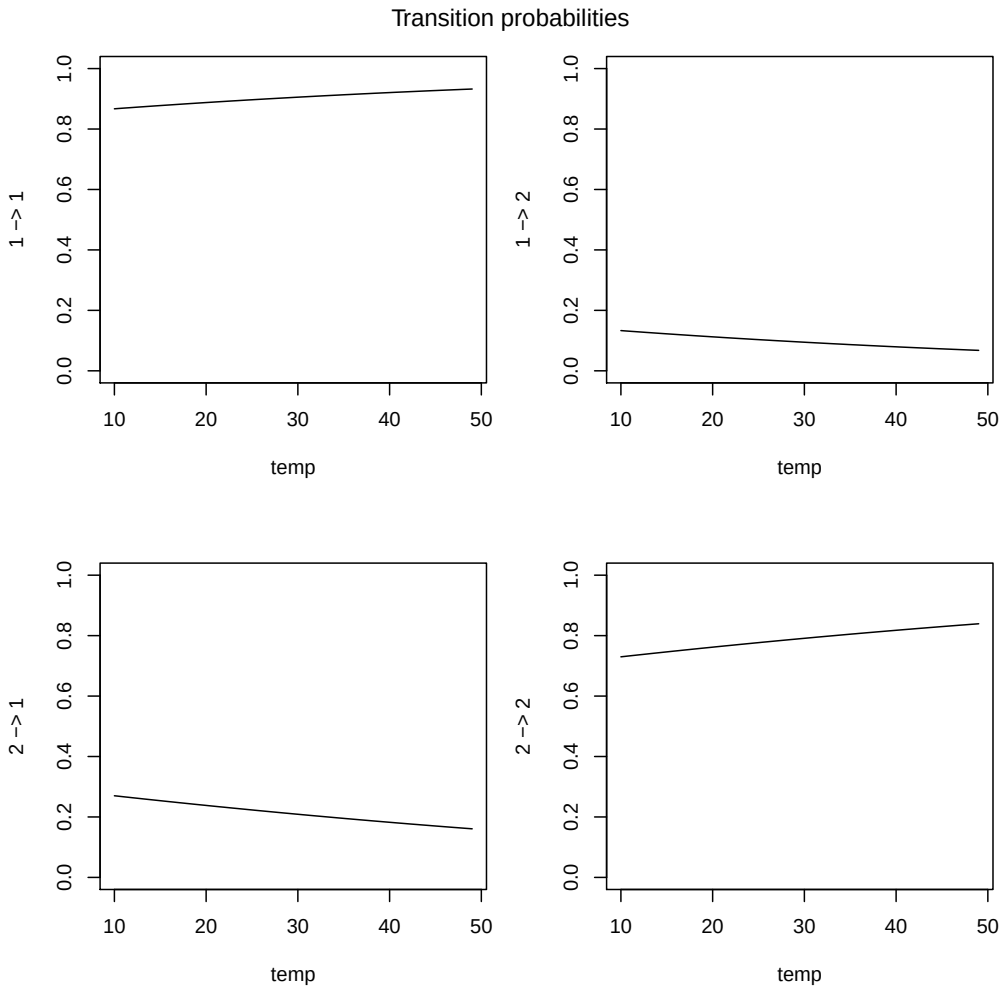
```
# temperature
mcov1 <- fitHMM(data, nbStates=2, stepPar0=c(stepMean0,stepSD0),
                angleMean0=c(angleMean0,angleCon0),
                formula=~temp)

# temperature and time of day, with interaction
mcov2 <- fitHMM(data, nbStates=2, stepPar0=c(stepMean0,stepSD0),
                angleMean0=c(angleMean0,angleCon0),
                formula=~temp*(sin(2*pi*tod/24)+cos(2*pi*tod/24)))
```

Covariates: temperature

```
mcov1


## Value of the maximum log-likelihood: -15408.82
##
## Step length parameters:
## ----------------------
##        state 1  state 2
## mean 0.2257244 1.213803
## sd   0.2411050 0.773870
##
## Turning angle parameters:
## ------------------------
##                   state 1     state 2
## mean           0.07096375 0.001745671
## concentration 0.56964288 2.474930789
##
## Regression coeffs for the transition probabilities:
## --------------------------------------------------
##                 1 -> 2      2 -> 1
## intercept -1.68198669 -0.82510791
## temp      -0.01922782 -0.01688585
##
## Initial distribution:
## --------------------
## [1] 0.9998178995 0.0001821005
```

Covariates: temperature

```
plot(mcov1)
```

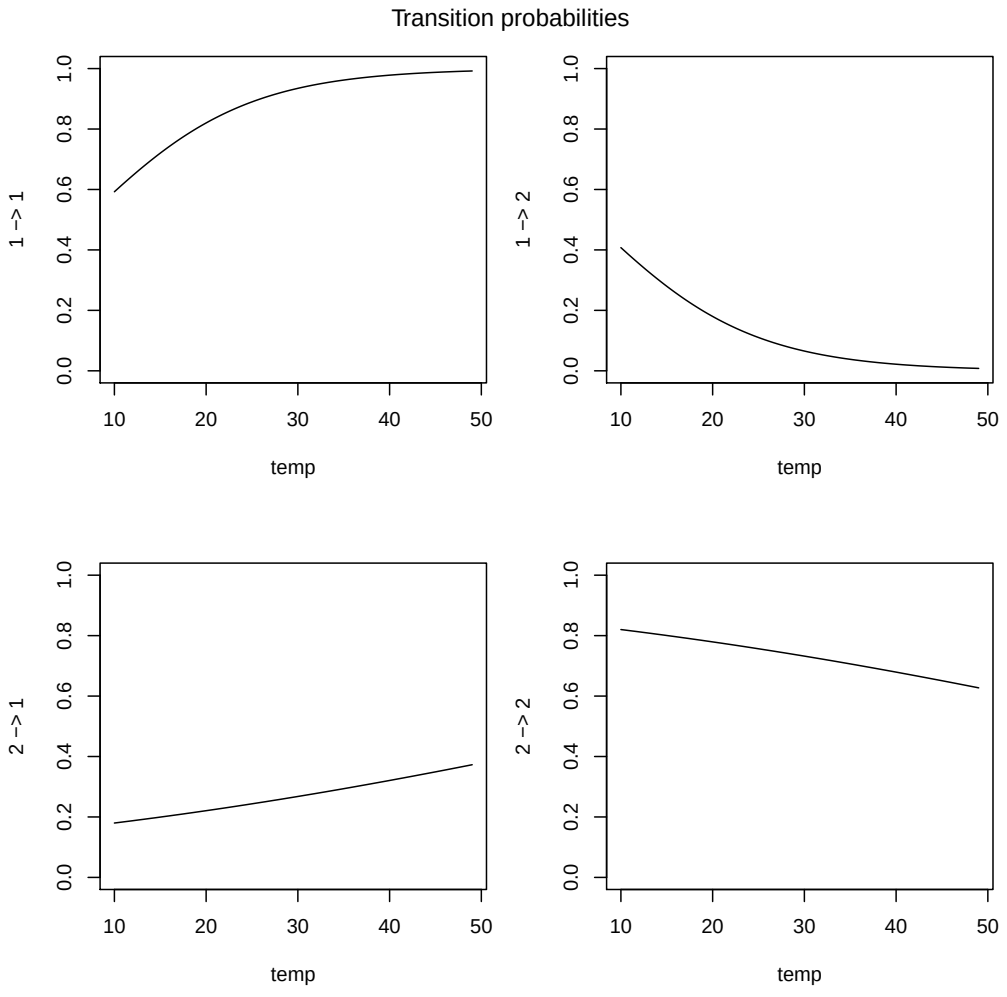


Covariates: temperature and time of day

```
mcov2


## Value of the maximum log-likelihood: -15284.4
##
## Step length parameters:
## ----------------------
##        state 1   state 2
## mean 0.2186661 1.1773090
## sd   0.2335630 0.7726656
##
## Turning angle parameters:
## ------------------------
##                   state 1      state 2
## mean          0.07693438 0.0003416462
## concentration 0.55010833 2.4137253090
##
## Regression coeffs for the transition probabilities:
## --------------------------------------------------
##                                1 -> 2      2 -> 1
## intercept                  -1.13598914 -0.86135961
## temp                       -0.03307267 -0.01599754
## sin(2 * pi * tod/24)        0.10732784 -0.28522334
## cos(2 * pi * tod/24)       -1.90758539  0.88145615
## temp:sin(2 * pi * tod/24) -0.03355243  0.02500032
## temp:cos(2 * pi * tod/24)  0.07754432 -0.03862041
##
## Initial distribution:
## --------------------
## [1] 9.999976e-01 2.395825e-06
```

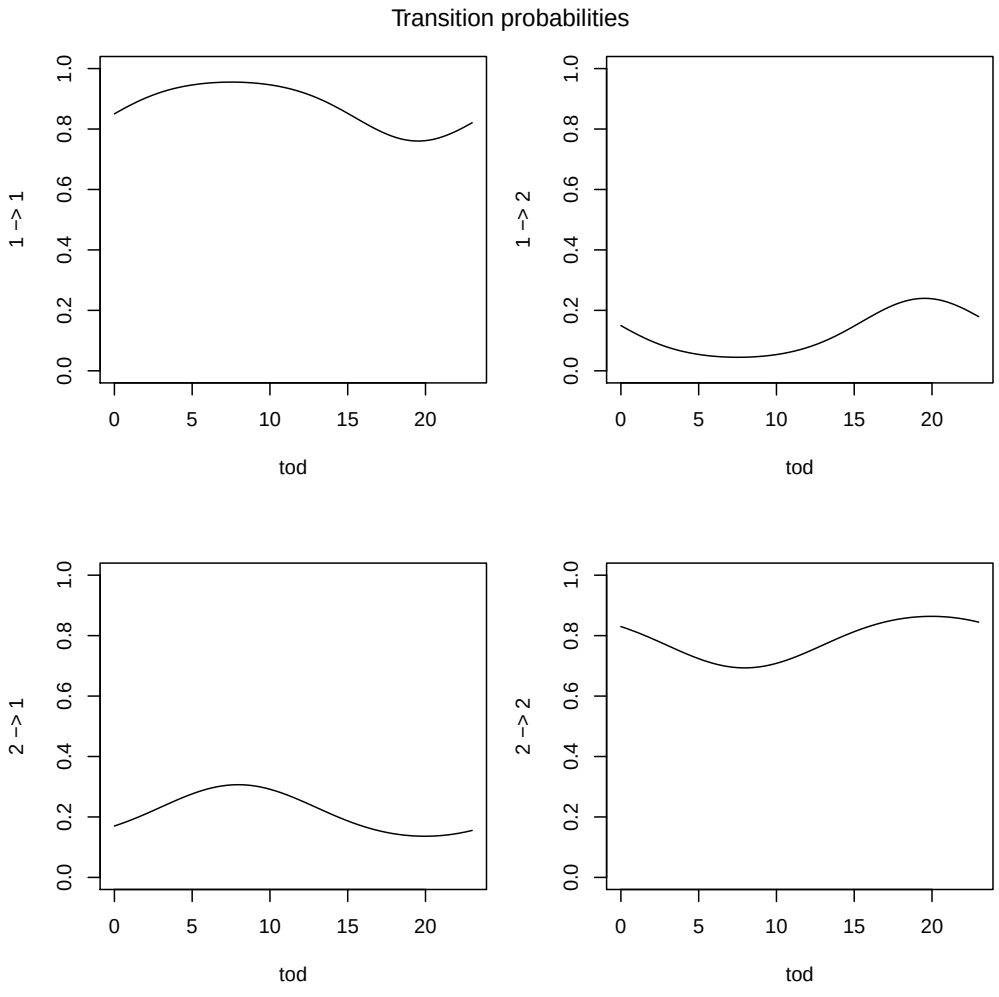
Covariates: temperature and time of day

```
plot(mcov2)
```



Covariates: temperature and time of day

```
plot(mcov2)
```



momentuHMM

Introduction to momentuHMM

- Extension of moveHMM, for more flexible models.
  $\longrightarrow$ Same core functions (prepData, fitHMM), with more options.
- Lead developer: Brett McClintock (NOAA).
- Available on CRAN since June 2017:

```
install.packages("momentuHMM")
```

To get started: the vignette presents seven case studies(!), to illustrate all the functionalitiesof the package.

Do I need moveHMM or momentuHMM?

moveHMM is easier to use, momentuHMM is more flexible.

Additional functionalities include:

- unlimited number of data streams;
- larger choice of distributions for data streams;
- covariates on the parameters of observation distributions;
- centres of interest/attraction;
- multiple imputation (irregular sampling, measurement error);
- ...

1. Elephant case study
2. Northern fur seal case study
3. Other examples

Fit a simple model with momentuHMM

```
data <- prepData(track, type="LL", covNames=c("temp","tod"))

# list of data streams
dist <- list(step="gamma", angle="vm")

# list of initial parameters
Par0 <- list(step=c(stepMean0,stepSD0), angle=c(angleCon0))

m1_mom <- fitHMM(data, nbStates=2, dist=dist, Par0=Par0)
```

Covariates (transition probabilities)

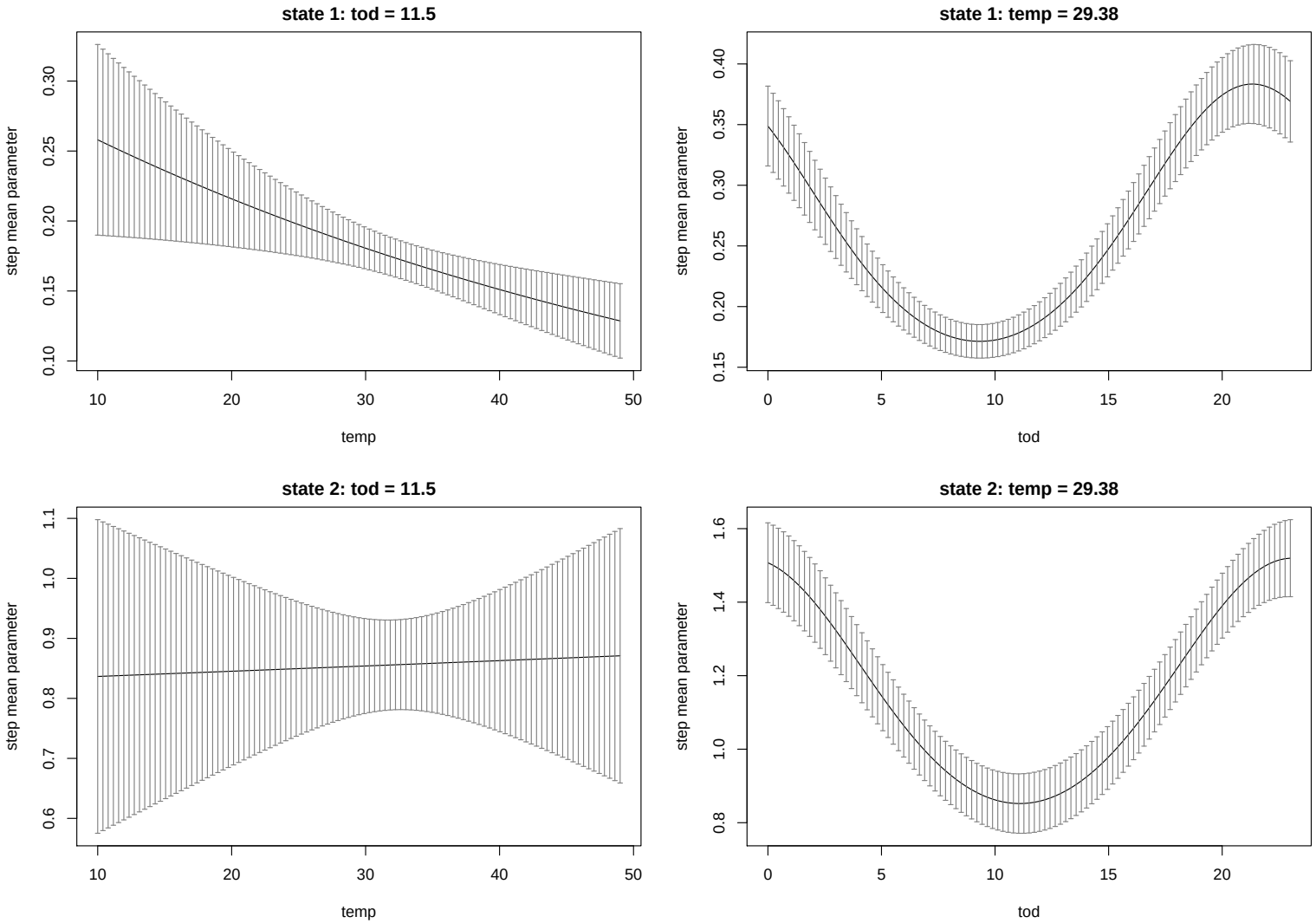
```
# formula for transition probabilities
formula <- ~temp*cosinor(tod, period=24)

# generate initial parameters for new model
Par0_m2 <- getPar0(model=m1_mom, formula=formula)

m2_mom <- fitHMM(data, nbStates=2, dist=dist, Par0=Par0_m2$Par,
                 beta0=Par0_m2$beta, formula=formula)
```

Covariates (step length distribution)

```
# formulas for observation parameters
DM <- list(step=list(mean=~temp*cosinor(tod,period=24),
                     sd=~temp*cosinor(tod,period=24)))

# generate initial parameters for new model
Par0_m3 <- getPar0(model=m2_mom, formula=formula, DM=DM)

m3_mom <- fitHMM(data, nbStates=2, dist=dist, Par0=Par0_m3$Par,
                 beta0=Par0_m3$beta, DM=DM, formula=formula)
```

Results – elephant model

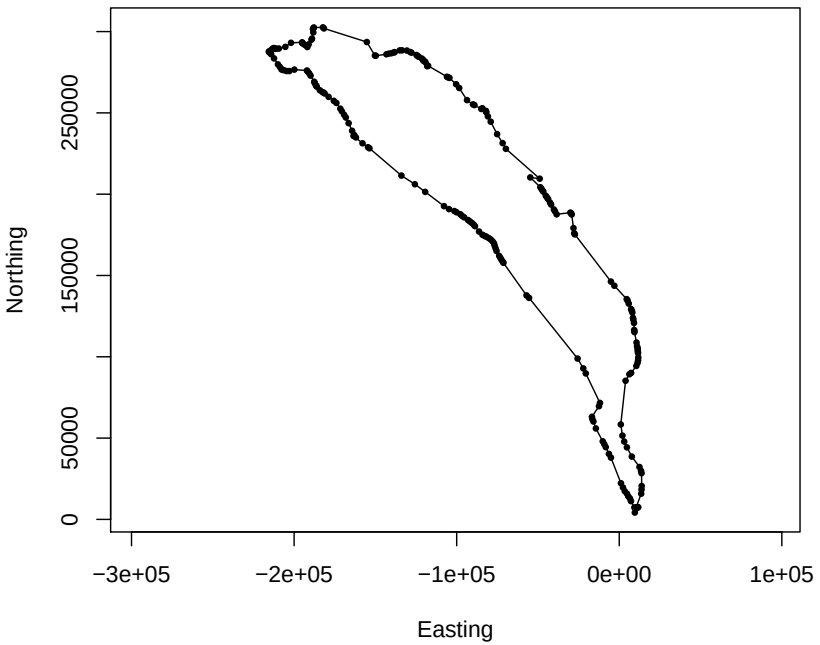
```
plot(m3_mom, plotCI=TRUE, ask=FALSE)
```

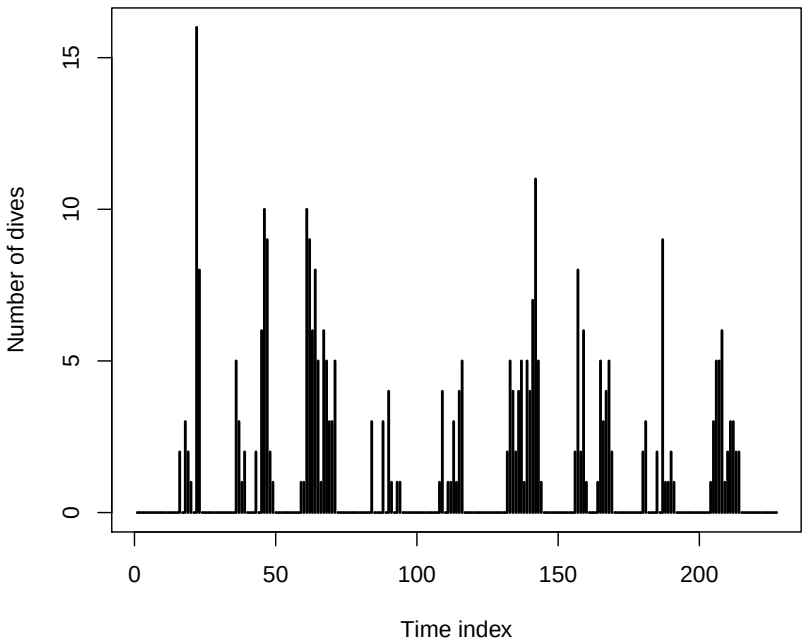


1. Elephant case study
2. Northern fur seal case study
3. Other examples

Northern fur seal data

- Locations (irregular sampling frequency)
- Number of dives per hour



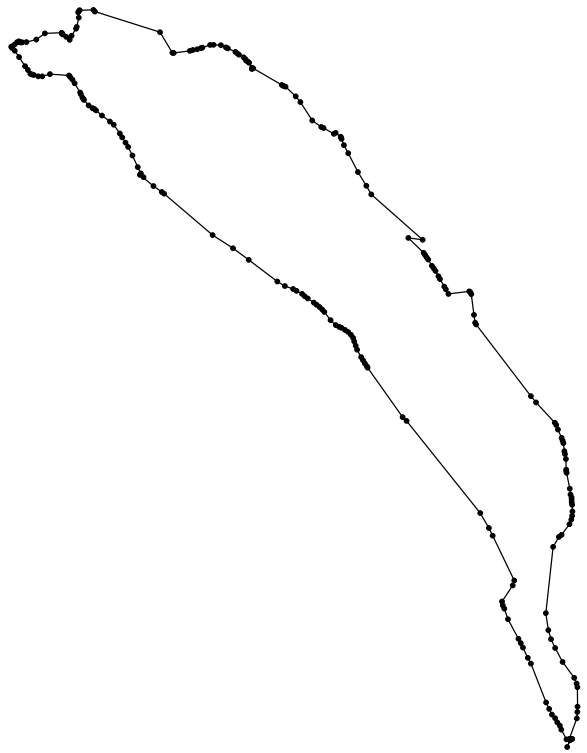


Data from McClintock et al. (2014), "When to be discrete: the importance of time formulation in understanding animal movement", Movement Ecology.

Multiple imputation

Solution to irregular sampling and measurement error:

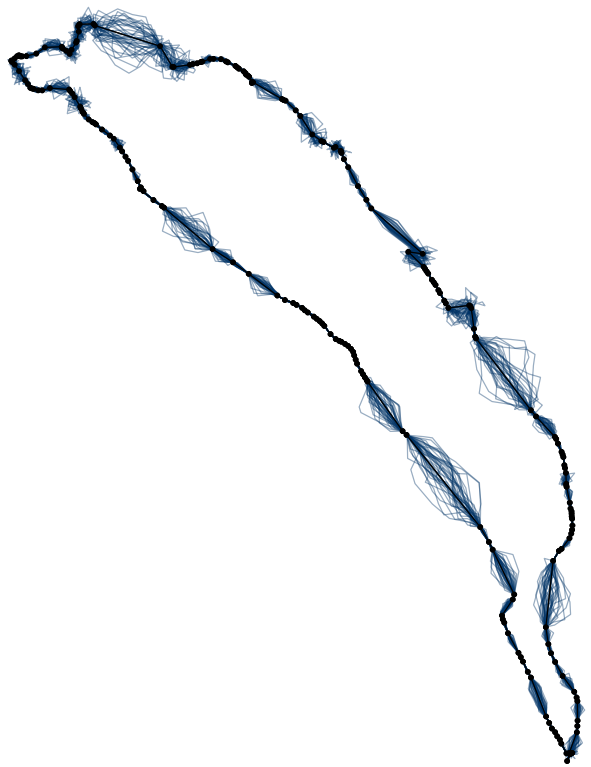
- Fit continuous-time movement model (crawl);
- Draw many regularly-sampled realisations from the fitted model;
- Fit a HMM to each realisation.



Multiple imputation

Solution to irregular sampling and measurement error:

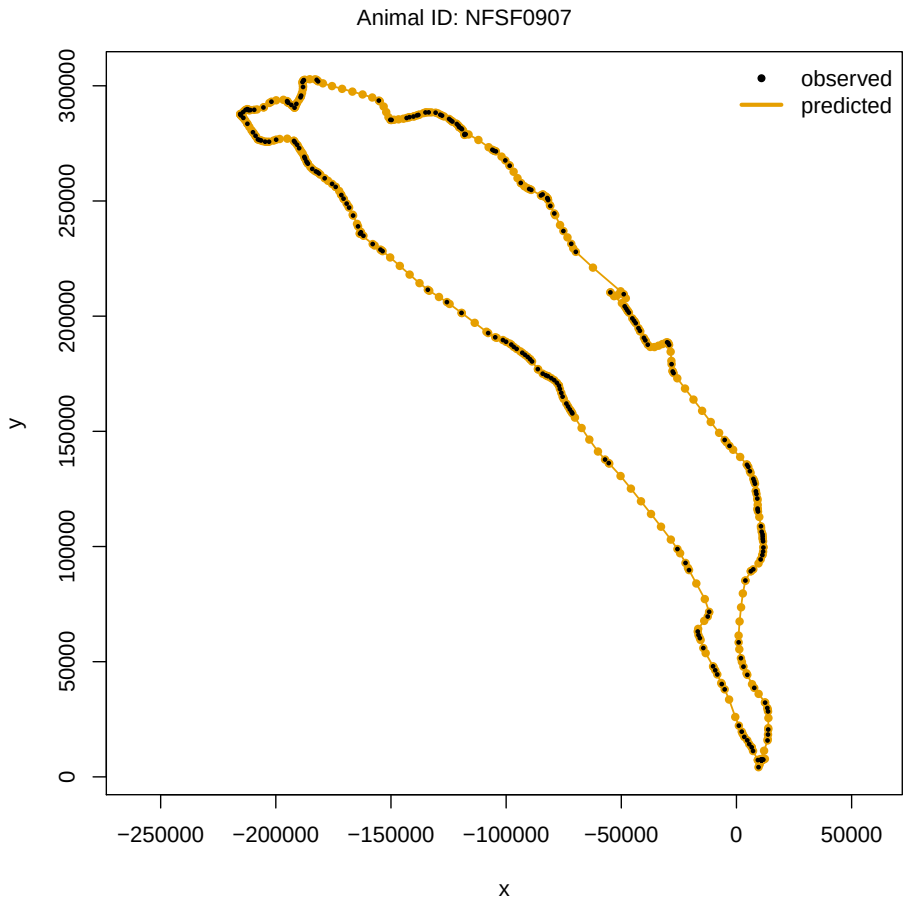
- Fit continuous-time movement model (crawl);
- Draw many regularly-sampled realisations from the fitted model;
- Fit a HMM to each realisation.



Fit crawl model

```
crwOut <- crawlWrap(obsData=nfsData, predTime=predTimes,
                    initial.state=inits,theta=c(4,-10),fixPar=c(NA,NA))
```

```
plot(crwOut, ask=FALSE)
```

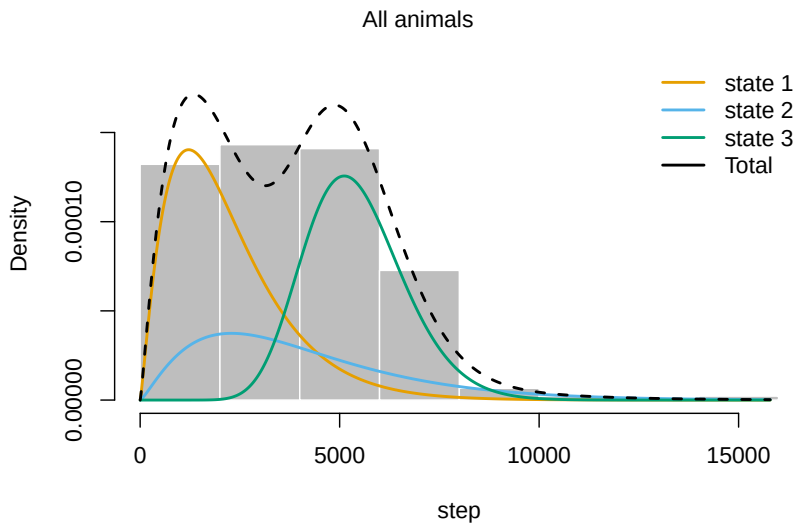


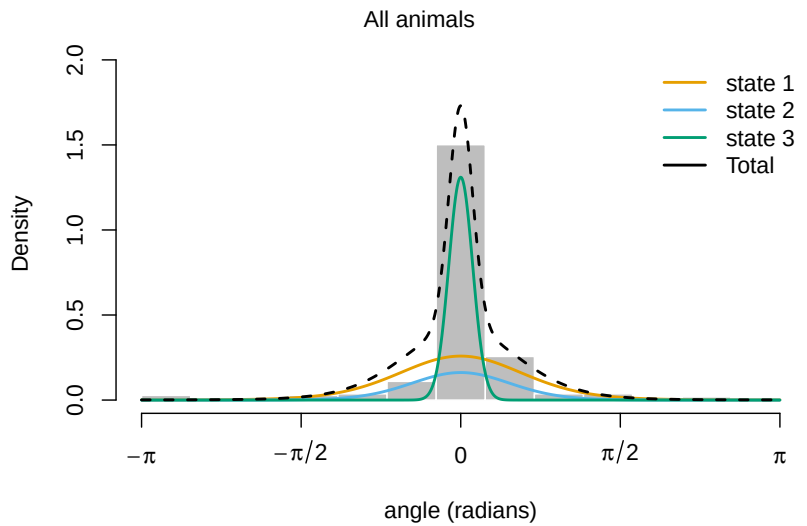
Fit HMM

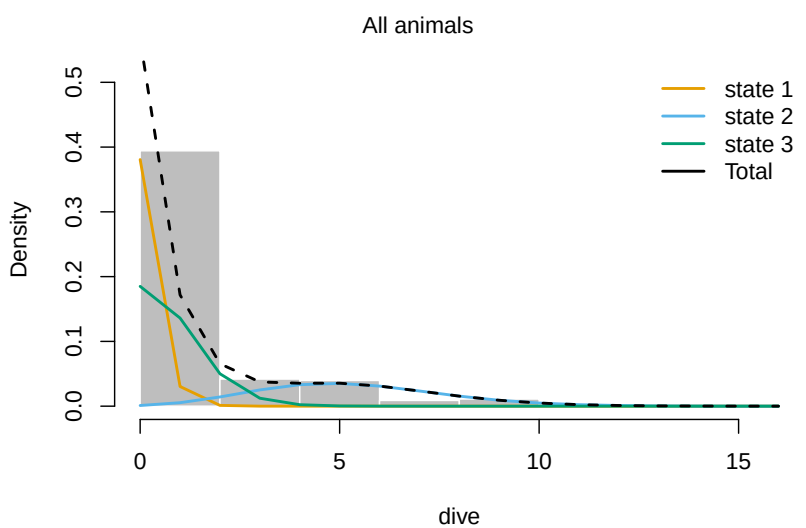
```r
# observation distributions
dist <- list(step="gamma", angle="vm", dive="pois")

# initial parameters
stepPar0 <- c(500,1000,5000,1000,1000,2000)
anglePar0 <- c(1,2,10)
divePar0 <- c(10e-4,2,10e-4)
Par0 <- list(step=stepPar0, angle=anglePar0, dive=divePar0)

# fit 3-state model
m_nfs <- fitHMM(data, nbStates=3, dist=dist, Par0=Par0)
```

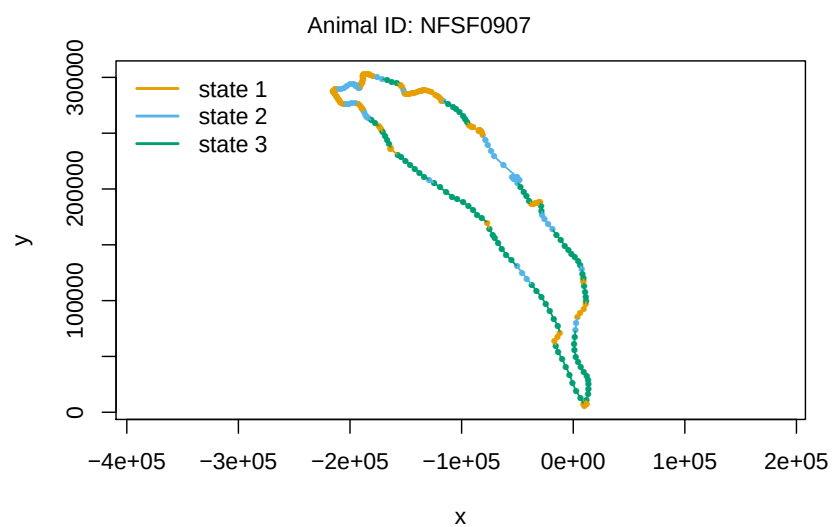
Results – NFS model

```
plot(m_nfs, ask=FALSE)
```



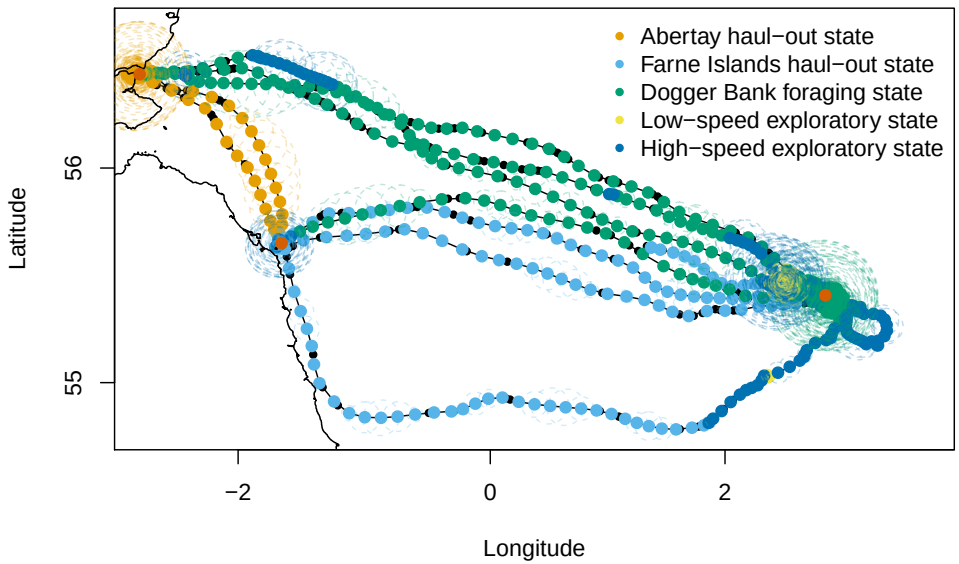


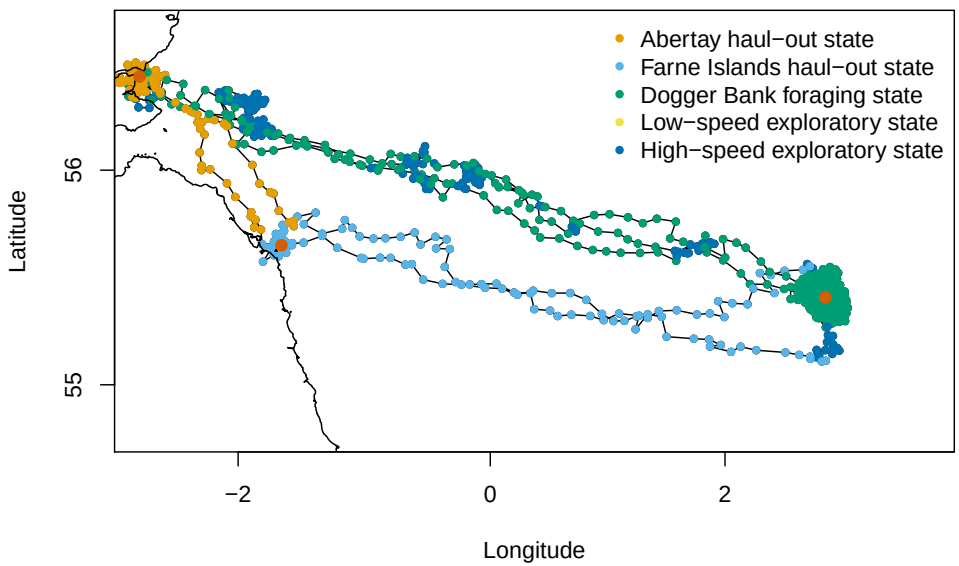




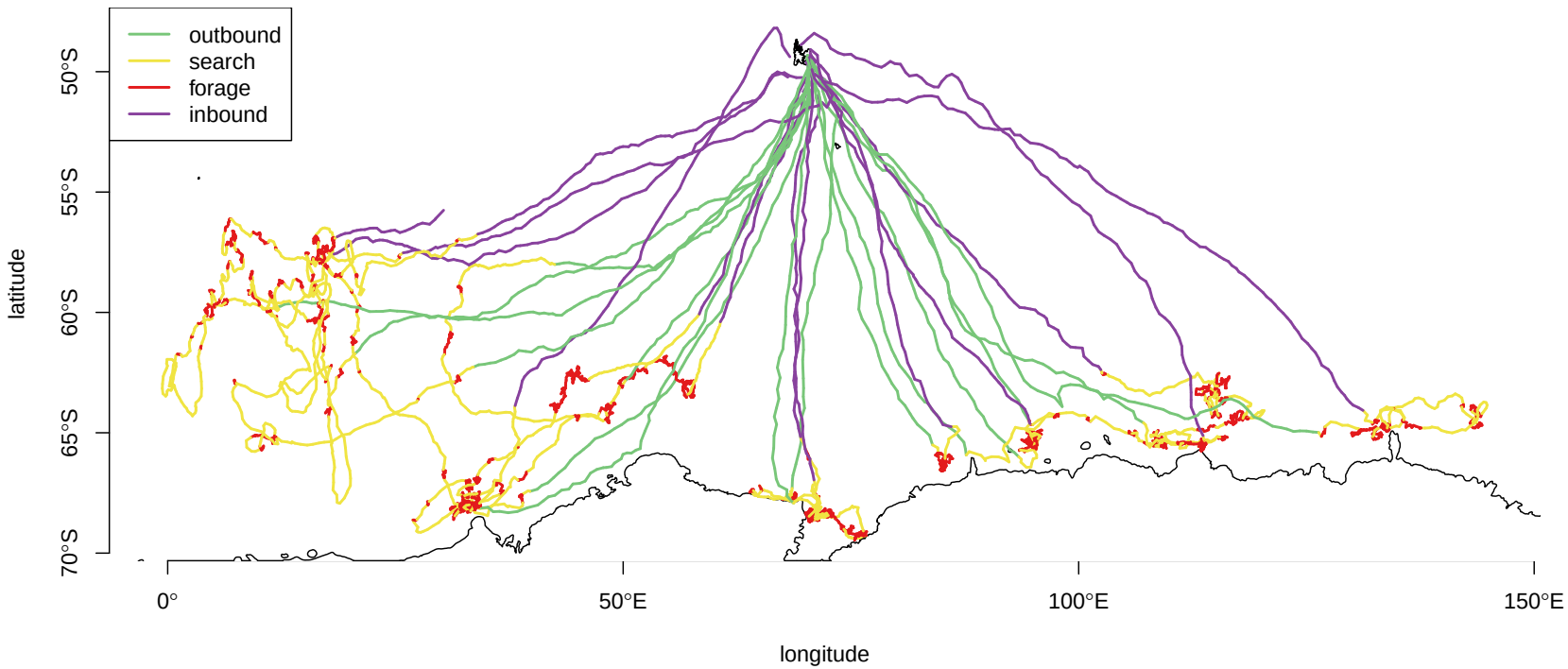
1. Elephant case study
2. Northern fur seal case study
3. Other examples

Grey seal


Abertay haul−out state
Farne Islands haul−out state
Dogger Bank foraging state
Low−speed exploratory state
High−speed exploratory state
Latitude
56
55
−2
0
2
Longitude


Abertay haul−out state
Farne Islands haul−out state
Dogger Bank foraging state
Low−speed exploratory state
High−speed exploratory state
Latitude
56
55
−2
0
2
Longitude

Elephant seals



Practical session