

moveHMM and momentuHMM:

R packages for inferring animal movement behaviour using multivariate HMMs

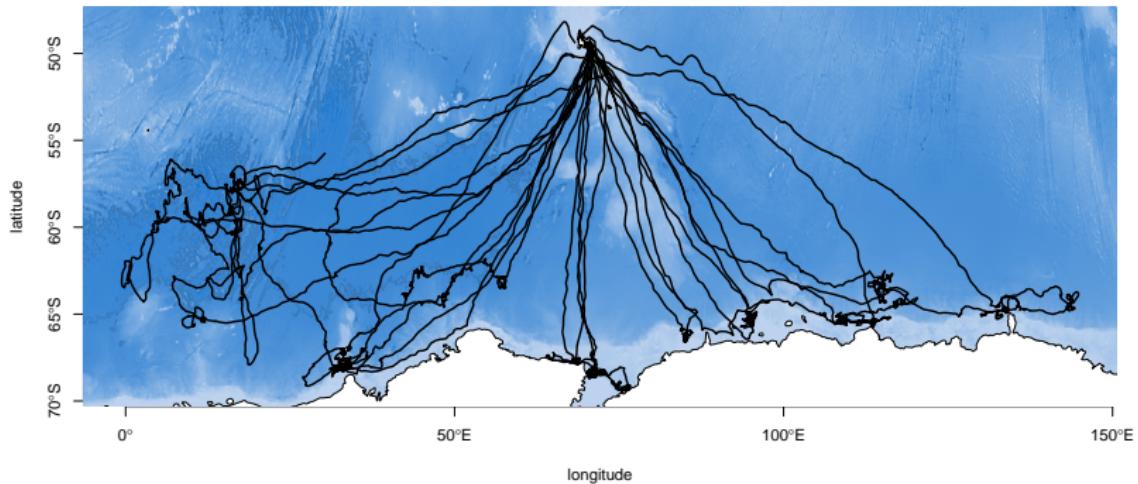
Brett T. McClintock¹, Théo Michelot²

¹*NOAA Marine Mammal Lab, Seattle, USA*

²University of St. Andrews

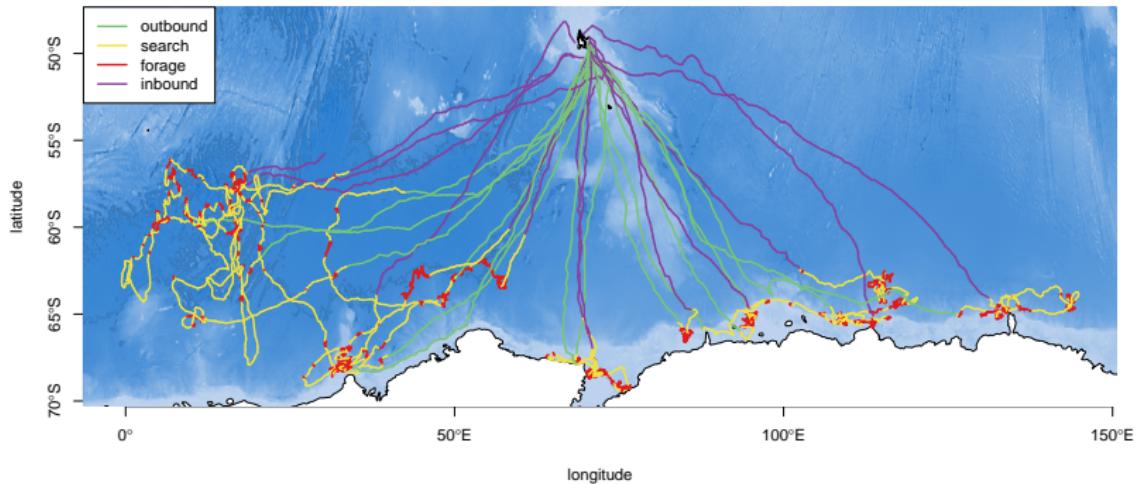
25 June 2022

HMM of animal movement



Data from: Michelot et al. (2017), "Estimation and simulation of foraging trips in land-based marine predators", *Ecology*, 98(7).

HMM of animal movement



Data from: Michelot et al. (2017), "Estimation and simulation of foraging trips in land-based marine predators", *Ecology*, 98(7).

Background on hidden Markov models

Correlated random walk

A correlated random walk includes **persistence in direction**.
→ Correlation between successive directions.

simple random walk



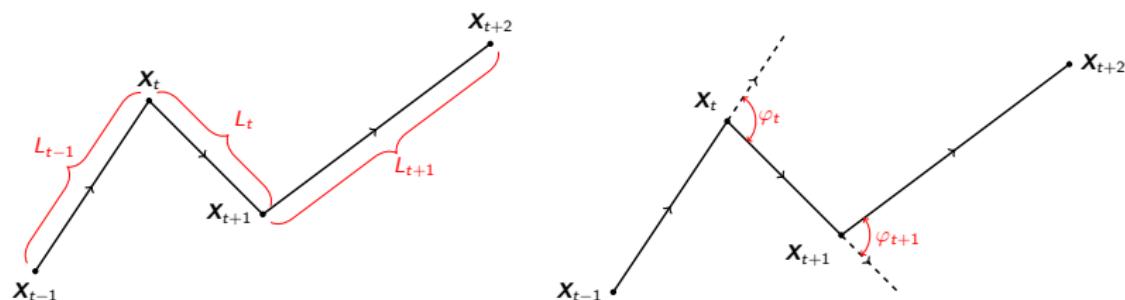
correlated random walk



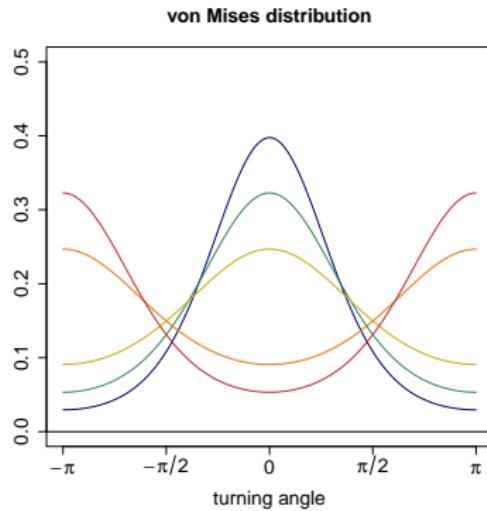
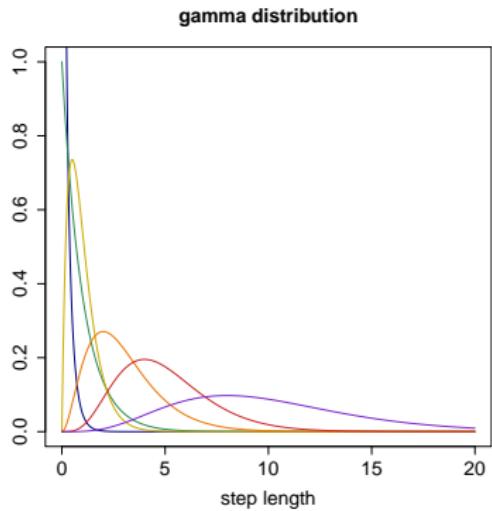
Movement metrics

In a correlated random walk, we can model:

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



Modelling the steps and angles



Multistate random walk

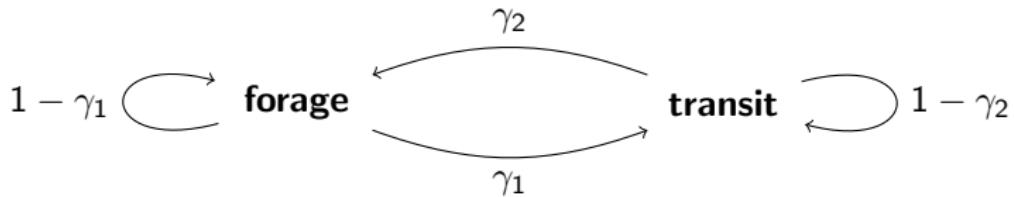
Idea: the animal switches between several movement processes, corresponding to several **behaviours**.
→ Behavioural process = unobserved Markov chain (S_t).

Multistate random walk

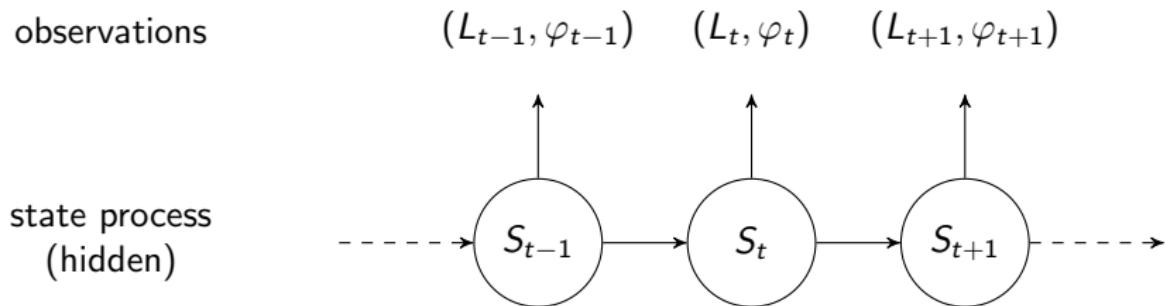
Idea: the animal switches between several movement processes, corresponding to several **behaviours**.

→ 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]?

→ 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

- ▶ Lead developer: Théo Michelot (University of St. Andrews)
- ▶ 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, *Methods Ecol Evol.*

moveHMM workflow

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

moveHMM overview: Elk example

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

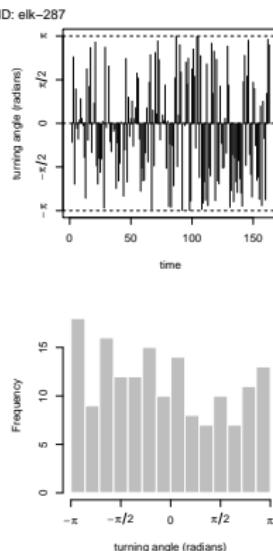
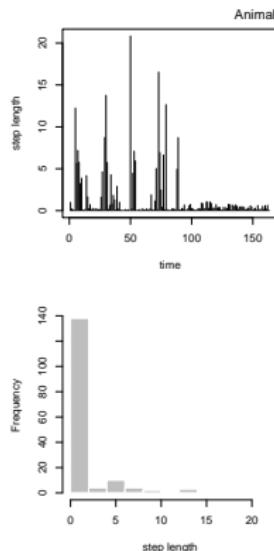
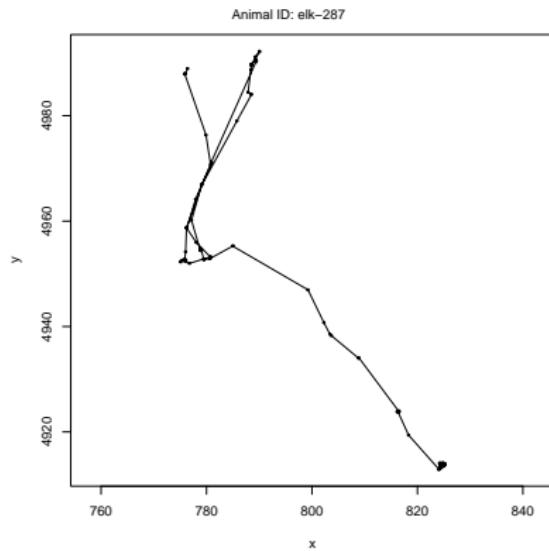
moveHMM overview: elk example

Prepare and visualize the data

```
trackData <- read.table("elk_data.txt", sep="\t", header=TRUE)

elkData <- prepData(trackData, type="UTM",
                     coordNames=c("Easting", "Northing"))

plot(elkData)
```



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

fitHMM

```
# define initial parameters
shape <- c(1,2) # step shape (two parameters: one for each state)
scale <- c(0.5,5) # step scale
zeromass0 <- c(0.01,0.001) # step zero-mass
stepPar0 <- c(shape,scale,zeromass0)
angleMean0 <- c(pi,0) # angle mean
rho <- c(0.2,0.75) # angle concentration
anglePar0 <- c(angleMean0,rho)

# fit 2-state model
doubleSwitch <- fitHMM(data = elkData,
                         nbStates = 2,
                         stepDist = "weibull",
                         angleDist = "wrpc Cauchy",
                         stepPar0, anglePar0)
```

Fitted model

doubleSwitch

```
## Value of the maximum log-likelihood: -1886.62
##
## Step length parameters:
## -----
##           state 1      state 2
## shape     0.835210568 1.368524e+00
## scale     0.399375912 6.431746e+00
## zero-mass 0.001618288 2.629002e-07
##
## Turning angle parameters:
## -----
##           state 1      state 2
## mean      -3.0664150 -0.001978805
## concentration 0.2619366  0.324342943
##
## Regression coeffs for the transition probabilities:
## -----
##           1 -> 2      2 -> 1
## intercept -2.350943 -0.0244189
##
## Transition probability matrix:
## -----
##           [,1]      [,2]
## [1,] 0.9130091 0.08699088
## [2,] 0.4938956 0.50610442
##
## Initial distribution:
## -----
## [1] 0.4047039 0.5952961
```

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(mod1,mod2,mod3)
 - ▶ 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.

```
doubleSwitch <- fitHMM(data = elkData,
                        nbStates = 2,
                        stepDist = "gamma",
                        angleDist = "vm",
                        stepPar0, anglePar0)
```

- ▶ Covariates on the switching probabilities.

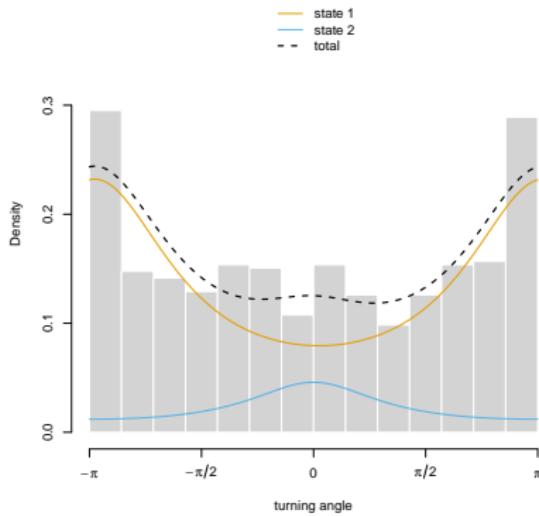
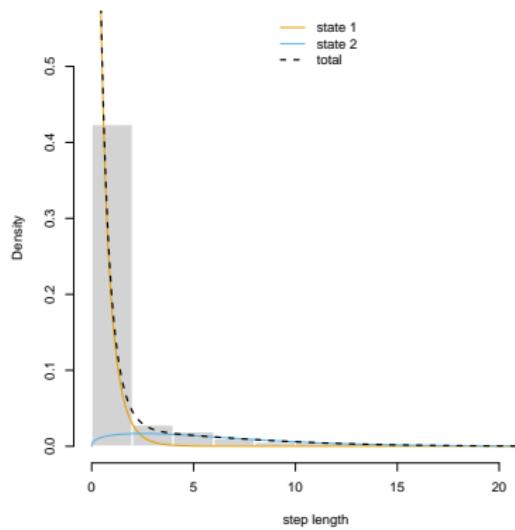
```
doubleSwitch <- fitHMM(data = elkData,
                        nbStates = 2,
                        stepDist = "weibull",
                        angleDist = "wrpc Cauchy",
                        stepPar0, anglePar0,
                        formula=~cov1+cov2)
```

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

Plot a fitted model

Visualize results

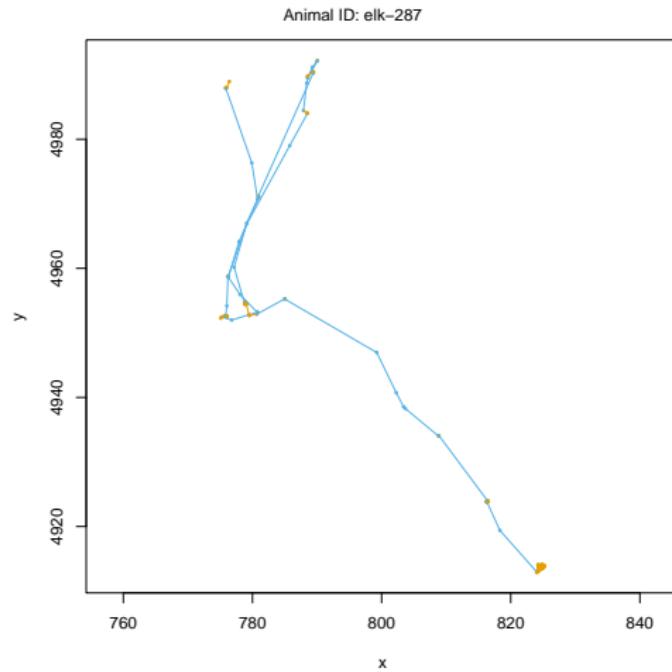
```
plot(doubleSwitch)
```



Plot a fitted model

Visualize results

```
plot(doubleSwitch)
```



Decode the state process

Visualize results

```
states <- viterbi(doubleSwitch)

head(states)

## [1] 2 2 1 1 1 2

sp <- stateProbs(doubleSwitch)

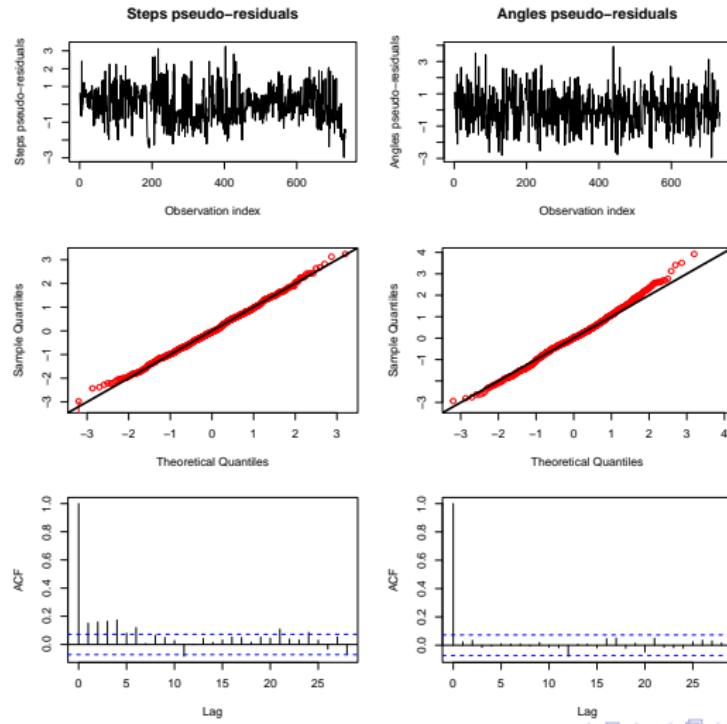
head(sp)

## [,1]      [,2]
## [1,] 9.392082e-04 0.99906079
## [2,] 3.233653e-01 0.67663473
## [3,] 9.899866e-01 0.01001337
## [4,] 9.865311e-01 0.01346893
## [5,] 9.774615e-01 0.02253846
## [6,] 1.185925e-05 0.99998814
```

Pseudo-residuals

Visualize results

```
plotPR(doubleSwitch)
```



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`

```
# habitat covariates
formula <- ~ dist_water + dist_swamp + dist_otw
      + dist_openfor + dist_ntw + dist_mixfor
      + dist-devel + dist_ddf + dist_conifer
      + dist_alvar

doubleSwitchCovs <- fitHMM(data=elkData, nbStates=2,
                           stepPar0=stepPar0, anglePar0=anglePar0,
                           stepDist="weibull", angleDist="wrpcalpha",
                           formula=formula)
```

Covariates: habitat type

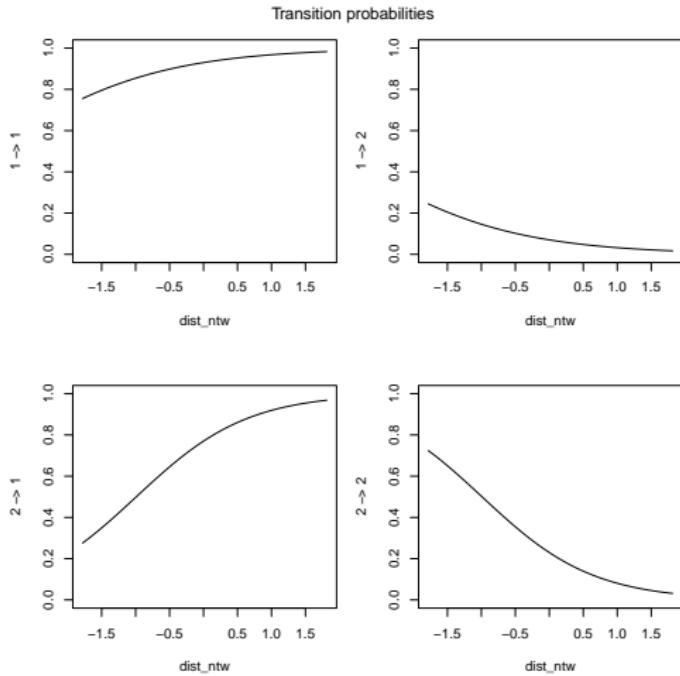
```
doubleSwitchCovs

## Value of the maximum log-likelihood: -1855.209
##
## Step length parameters:
## -----
##           state 1      state 2
## shape     0.841149467 1.422552e+00
## scale     0.397301348 6.568901e+00
## zero-mass 0.001615238 8.080851e-08
##
## Turning angle parameters:
## -----
##           state 1      state 2
## mean      -3.0692720 -0.0401013
## concentration 0.2590377 0.3126627
##
## Regression coeffs for the transition probabilities:
## -----
##           1 -> 2      2 -> 1
## intercept   -2.5852968  1.20963393
## dist_water   -0.2415109  0.01955078
## dist_swamp    0.9332552 -1.30407980
## dist_otw     -0.4439618  0.14718842
## dist_openfor  -1.2749267  1.59483590
## dist_ntw     -0.8184189  1.22154805
## dist_mixfor   -0.0175164 -0.32198948
## dist-devel    -0.7534484  1.31687360
## dist_ddf      0.4419557 -0.26185725
## dist_conifer   0.1418536  0.33904707
## dist_alvar    -0.3881678  0.23654766
##
## Initial distribution:
## -----
```



Covariates: habitat type

```
plot(doubleSwitchCovs)
```



Model selection

```
AIC(doubleSwitch, doubleSwitchCovs)
```

```
##                 Model      AIC
## 1 doubleSwitchCovs 3776.419
## 2      doubleSwitch 3799.241
```

momentuHMM

Motivation

- ▶ Large amounts of complex high-resolution telemetry data:
 - ▶ location
 - ▶ depth or altitude
 - ▶ acceleration, stomach temperature...
- ▶ Large amounts of spatio-temporal environmental data
 - ▶ habitat type
 - ▶ temperature
 - ▶ ocean currents, wind velocity
 - ▶ sea ice concentration...
- ▶ How to learn about the animal's behaviour?
- ▶ How to link movement and behaviour to covariates?

Motivation

- ▶ Large amounts of complex high-resolution telemetry data:
 - ▶ location
 - ▶ depth or altitude
 - ▶ acceleration, stomach temperature...
- ▶ Large amounts of spatio-temporal environmental data
 - ▶ habitat type
 - ▶ temperature
 - ▶ ocean currents, wind velocity
 - ▶ sea ice concentration...
- ▶ How to learn about the animal's behaviour?
- ▶ How to link movement and behaviour to covariates?

→ **Generalized multivariate hidden Markov models** ←

Introduction to momentuHMM

- ▶ Extension of moveHMM, for more flexible models
→ Same core functions with more options
- ▶ Available on CRAN:

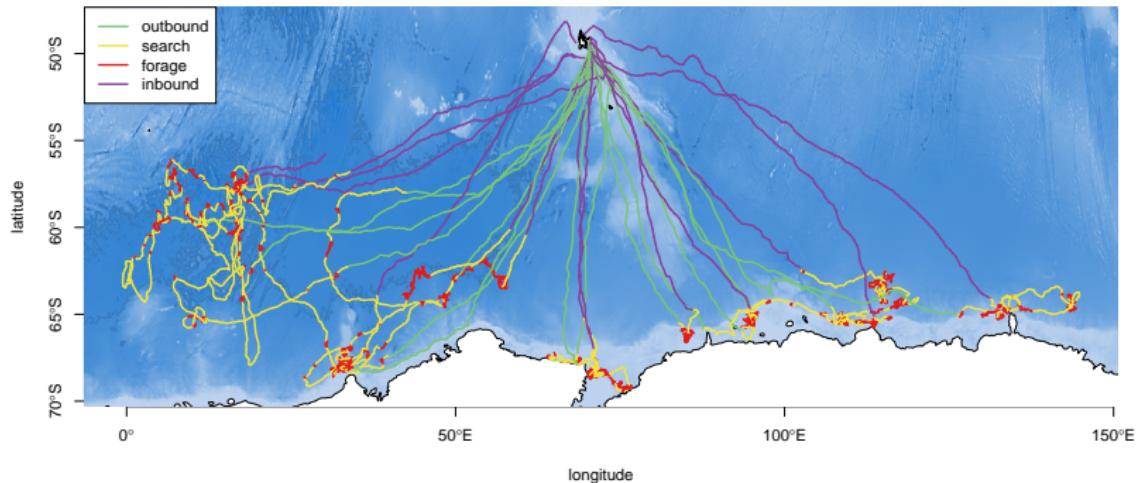
```
install.packages("momentuHMM")
```

To get started: the vignette presents 15 case studies to illustrate all the functionalities of the package.

-  McClintock and Michelot (2018). momentuHMM: R package for generalized hidden Markov models of animal movement, *Methods Ecol Evol.*

Do I need moveHMM or momentuHMM?

moveHMM is easier to use, momentuHMM is much more flexible



Data from: Michelot et al. (2017), "Estimation and simulation of foraging trips in land-based marine predators", *Ecology*, 98(7).

Do I need moveHMM or momentuHMM?

Additional capabilities of momentuHMM include:

- ▶ biased and correlated random walks
- ▶ unlimited number of data streams
- ▶ larger choice of distributions for data streams
- ▶ covariates on all parameters
- ▶ parameter constraints
- ▶ seamless integration of spatio-temporal covariate raster data
- ▶ centres of attraction, potential functions
- ▶ group dynamic models
- ▶ cosinor (i.e. cyclic) and spline models for complex patterns
- ▶ multiple imputation (irregular sampling, measurement error)
- ▶ parallel processing
- ▶ ...

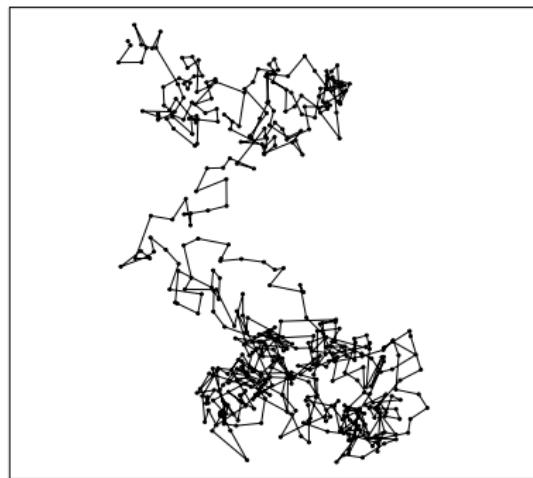
Biased random walk

A biased random walk includes **bias (or drift) in direction**

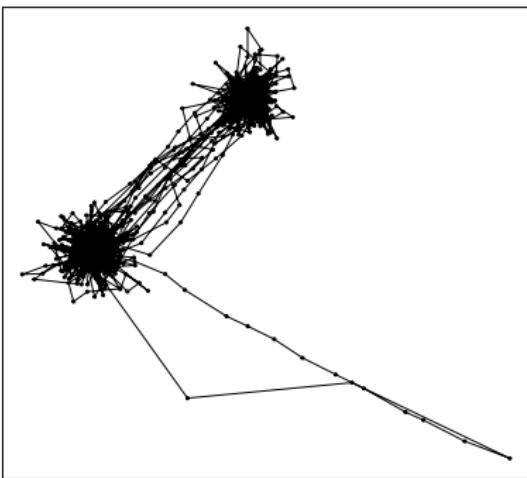
→ along gradients (e.g. ocean currents, wind velocity)

→ towards (or away from) areas of attraction (or repulsion)

simple random walk

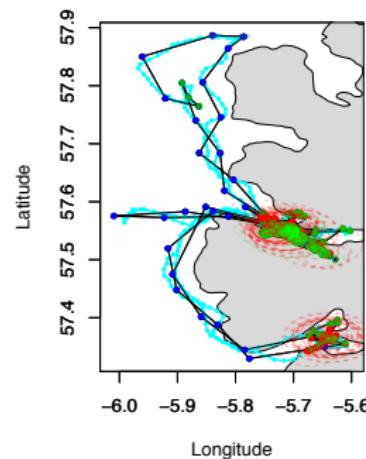
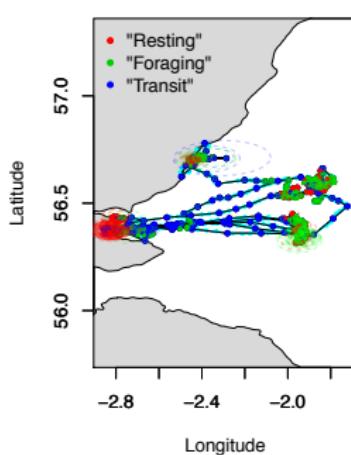


biased random walk

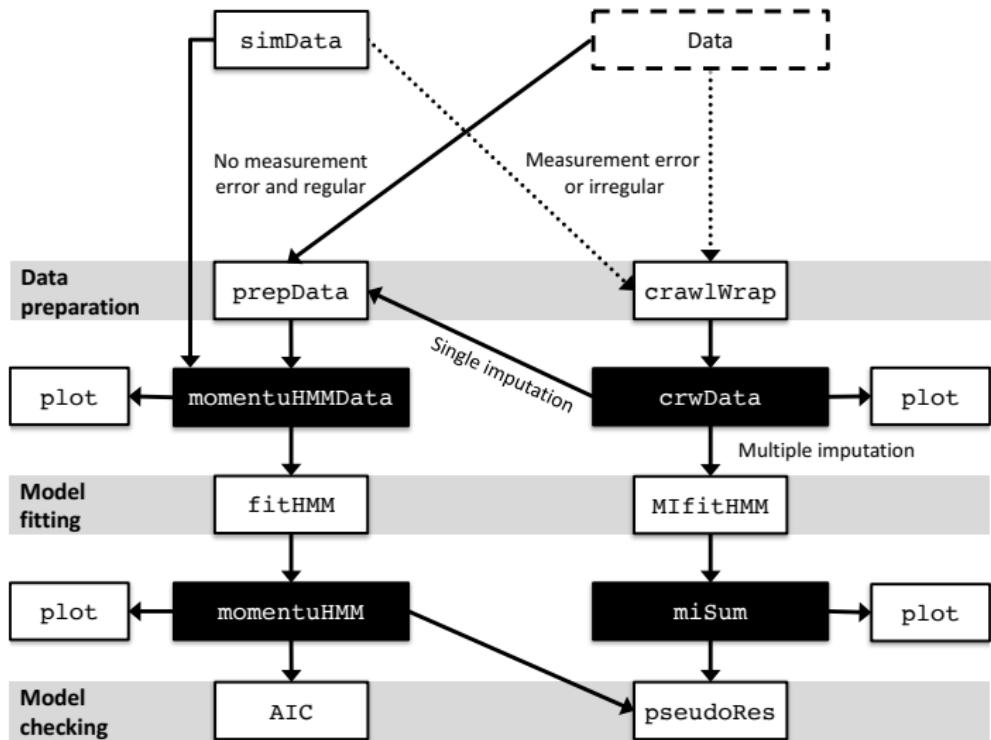


Auxiliary biotelemetry and environmental data

- ▶ Often difficult to discern > 2 behaviour states that are biologically meaningful based solely on horizontal trajectory
- ▶ Additional data streams can help distinguish states that would otherwise be difficult or impossible to discern
- ▶ e.g., proportion of each time step spent diving allows 3 behaviour states to be identified in marine mammals:



momentuHMM workflow

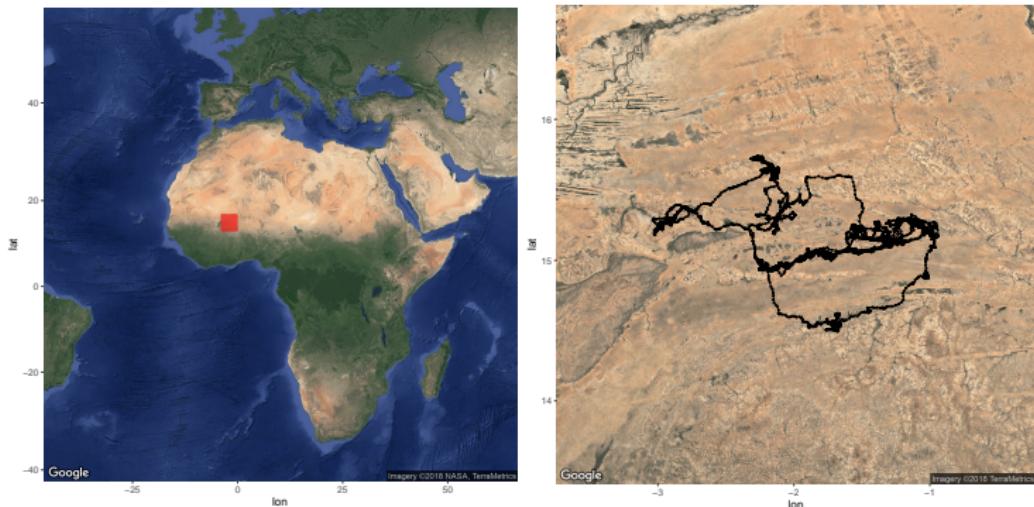


Schematic representing the typical `momentuHMM` workflow. White boxes indicate package functions and black boxes indicate object classes returned by functions.

1. Elephant example
2. Northern fur seal example
3. Other examples

Elephant example

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.)

Formatting the data

```
track <- read.csv("elephant.csv")  
  
head(track)  
  
##      ID          x          y temp tod  
## 1  -2.160167 15.65350   38  17  
## 2  -2.160075 15.65452   35  18  
## 3  -2.159902 15.65451   32  19  
## 4  -2.159435 15.65489   30  20  
## 5  -2.158113 15.65512   29  21  
## 6  -2.157848 15.65461   28  22
```

prepData

```
library(momentuHMM)

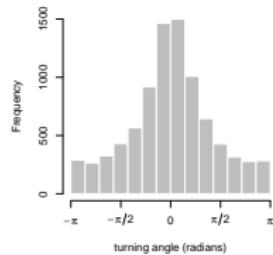
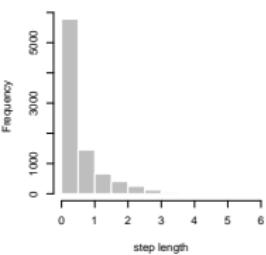
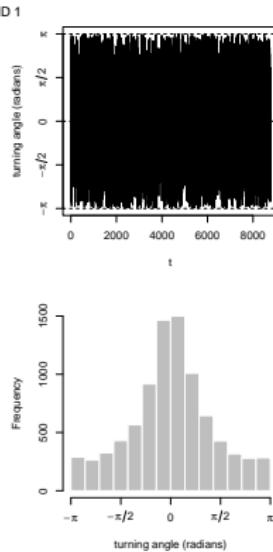
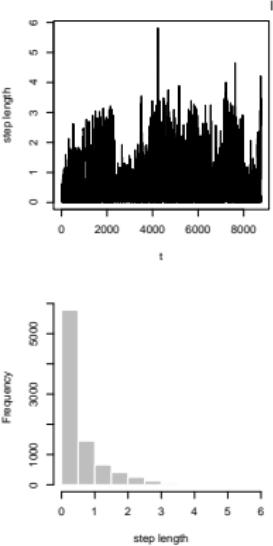
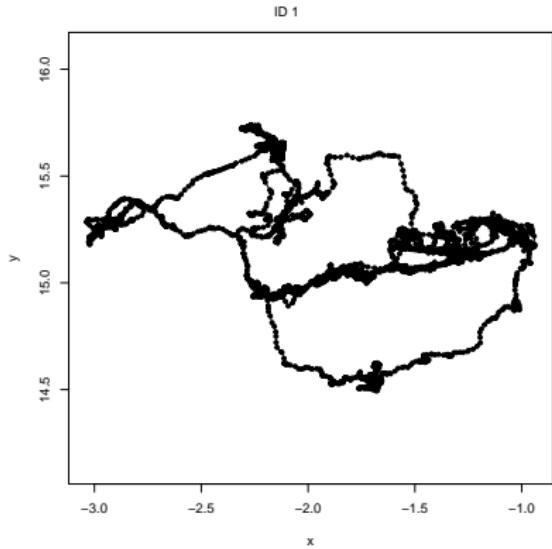
data <- prepData(track, type="LL", covNames=c("temp","tod"))

head(data)

##      ID      step     angle        x        y temp tod
## 1 1 0.11329501       NA -2.160167 15.65350  38 17
## 2 1 0.01862565 -1.5534720 -2.160075 15.65452  35 18
## 3 1 0.06477228  0.7575710 -2.159902 15.65451  32 19
## 4 1 0.14400817 -0.5090122 -2.159435 15.65489  30 20
## 5 1 0.06253090 -1.2781268 -2.158113 15.65512  29 21
## 6 1 0.20910447  2.8261263 -2.157848 15.65461  28 22
```

Visualise the data

```
plot(data)
```



Fit the basic 2-state model

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

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

m1 <- fitHMM(data, nbStates=2, dist=dist, Par0=Par0)

## -----
## Fitting HMM with 2 states and 2 data streams
##
## step ~ gamma(mean=~1, sd=~1)
## angle ~ vm(concentration=~1)
##
## Transition probability matrix formula: ~1
##
## Initial distribution formula: ~1
##
## DONE
```

Fitted model

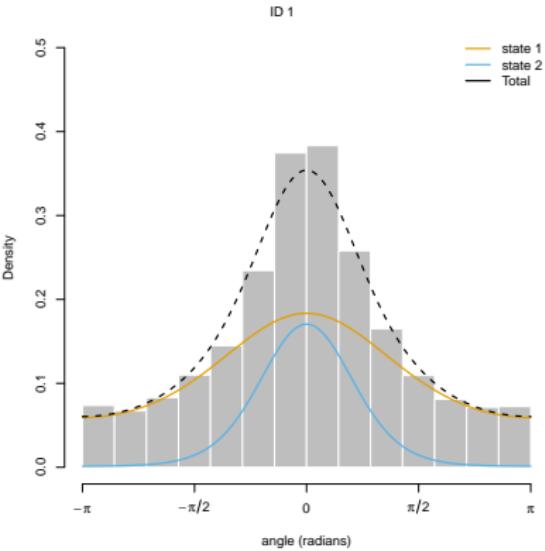
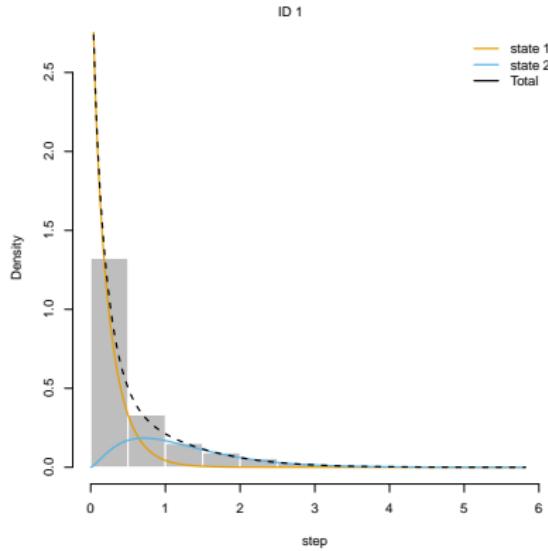
m1

```
## Value of the maximum log-likelihood: -15414.76
##
## step parameters:
## -----
##           state 1   state 2
## mean  0.2248231 1.2082076
## sd    0.2401735 0.7743595
##
## angle parameters:
## -----
##           state 1   state 2
## mean      0.0000000 0.000000
## concentration 0.5651901 2.467025
##
## Regression coeffs for the transition probabilities:
## -----
##           1 -> 2   2 -> 1
## (Intercept) -2.245593 -1.299832
##
## Transition probability matrix:
## -----
##           state 1   state 2
## state 1 0.9042697 0.09573031
## state 2 0.2141933 0.78580675
##
## Initial distribution:
## -----
##           state 1       state 2
## 9.999855e-01 1.451539e-05
```



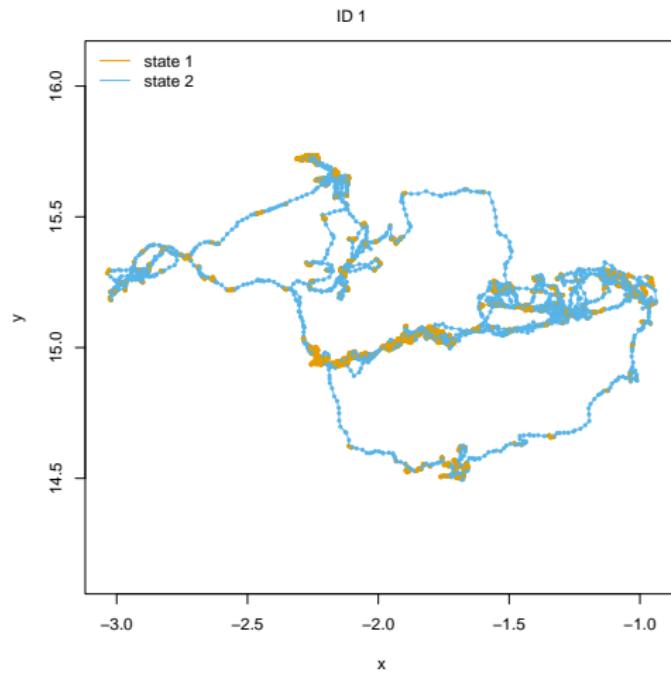
Plot a fitted model

`plot(m1)`



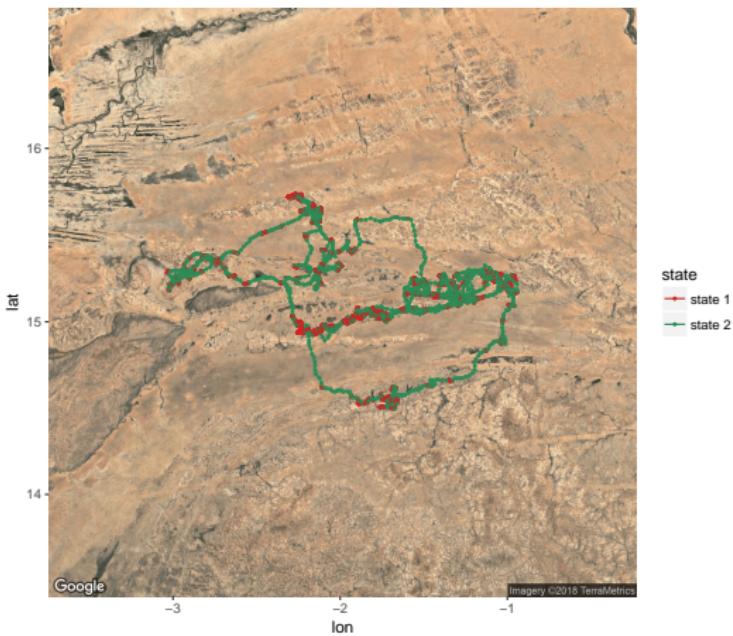
Plot a fitted model

plot(m1)



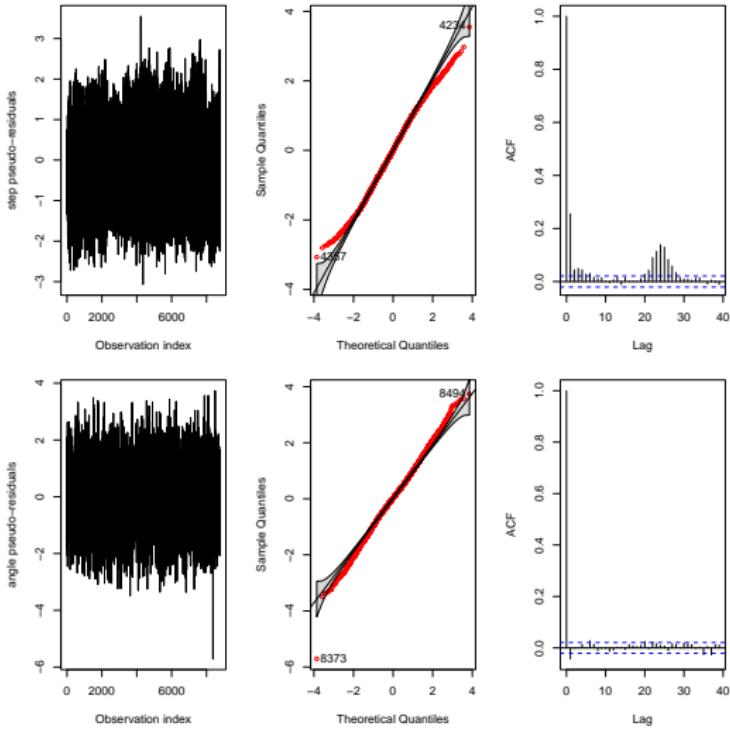
Plot the decoded track

```
plotSat(m1, zoom=8,  
        col = c("firebrick3","seagreen4"))
```



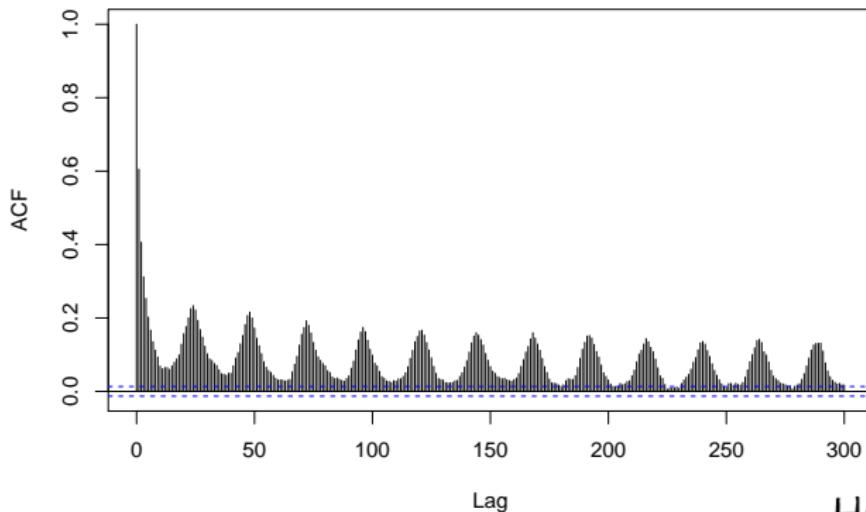
Pseudo-residuals

plotPR(m1)



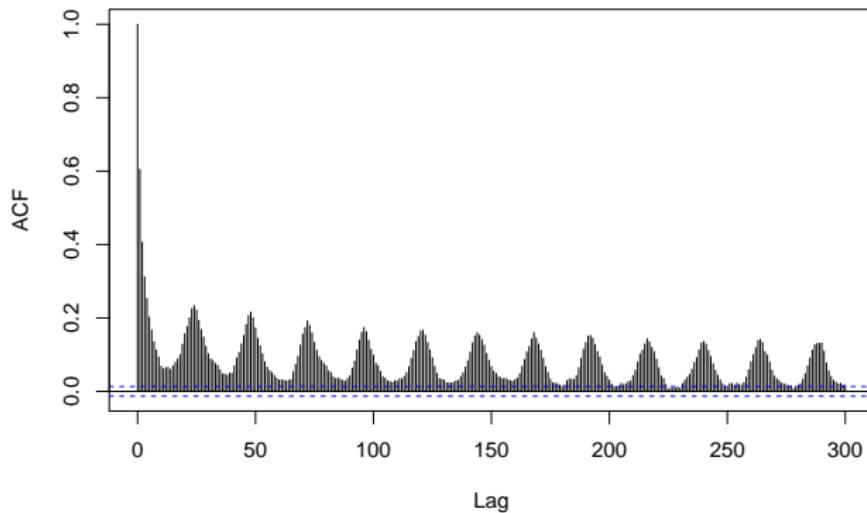
Let's look deeper...

```
acf(data$step, lag.max=300)
```



Let's look deeper...

```
acf(data$step, lag.max=300)
```



→ ACF plot highly suggests 24-hour periodicity in step lengths

Covariates (transition probabilities)

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

# generate initial parameters for new model
Par0_m2 <- getPar0(model=m1, formula=formula)

m2 <- fitHMM(data, nbStates=2, dist=dist, Par0=Par0_m2$Par,
               beta0=Par0_m2$beta, formula=formula)
```

```
## =====
## Fitting HMM with 2 states and 2 data streams
## -----
## step ~ gamma(mean=~1, sd=~1)
## angle ~ um(concentration=~1)
##
## Transition probability matrix formula: ~temp * cosinor(tod, period = 24)
## 
## Initial distribution formula: ~1
## =====
```

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, formula=formula, DM=DM)

m3 <- fitHMM(data, nbStates=2, dist=dist, Par0=Par0_m3$Par,
               beta0=Par0_m3$beta, DM=DM, formula=formula)
```

```
## =====
## Fitting HMM with 2 states and 2 data streams
## -----
## step ~ gamma(mean=~temp * cosinor(tod, period = 24), sd=~temp * cosinor(tod, period = 24))
## angle ~ um(concentration=~1)
## 
## Transition probability matrix formula: ~temp * cosinor(tod, period = 24)
## 
## Initial distribution formula: ~1
## =====
```

Model selection

```
AIC(m1,m2,m3)
```

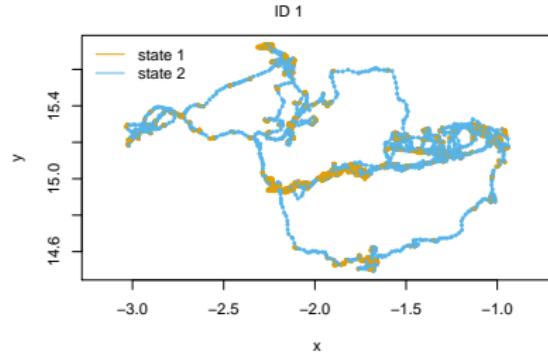
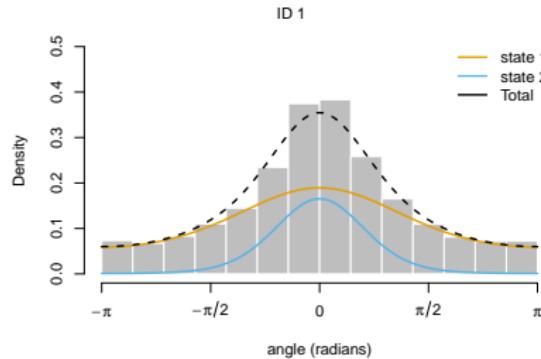
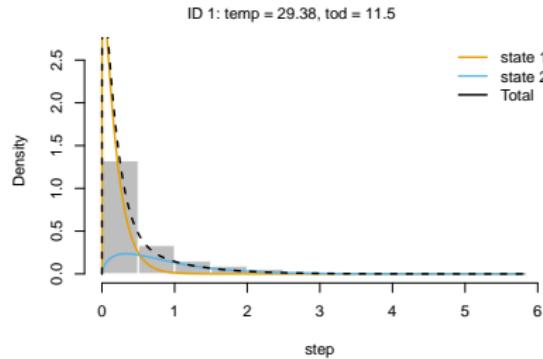
```
##      Model      AIC
## 1      m3 30347.87
## 2      m2 30612.00
## 3      m1 30847.51
```

```
AICweights(m1,m2,m3)
```

```
##      Model      weight
## 1      m3 1.000000e+00
## 2      m2 4.412220e-58
## 3      m1 3.192155e-109
```

Results – elephant model

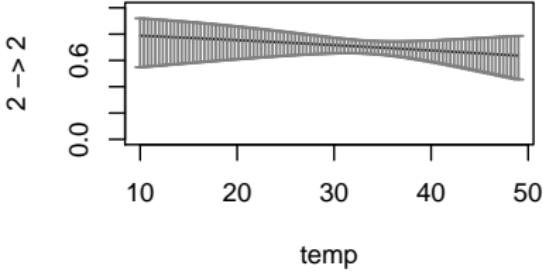
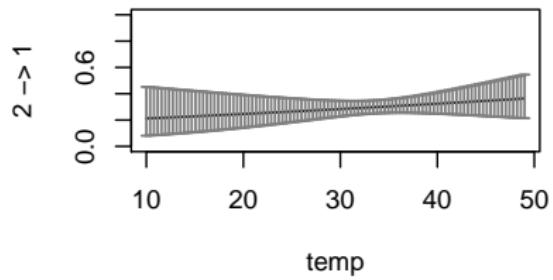
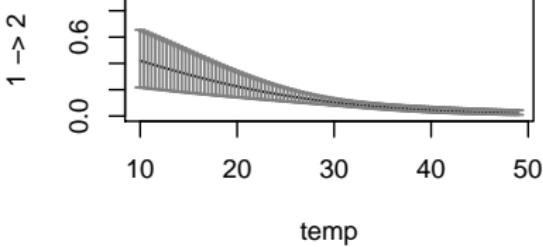
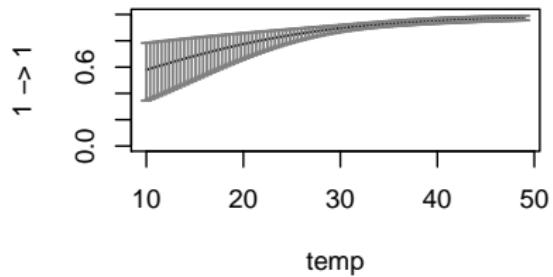
plot(m3)



Covariates: temperature and time of day

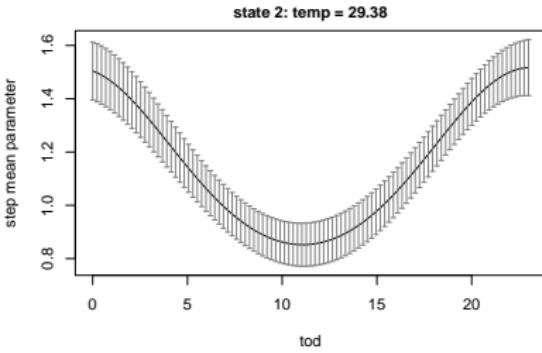
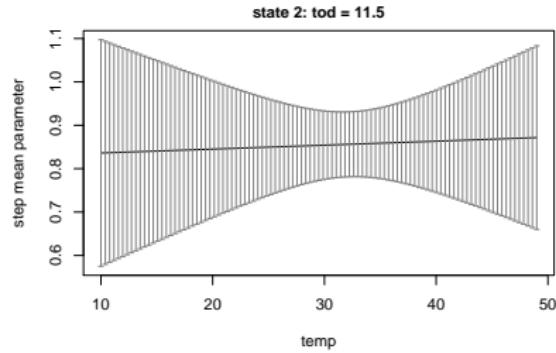
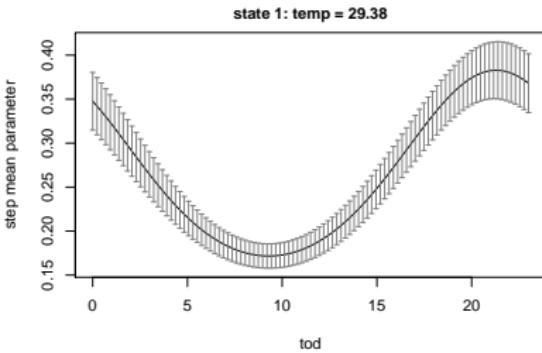
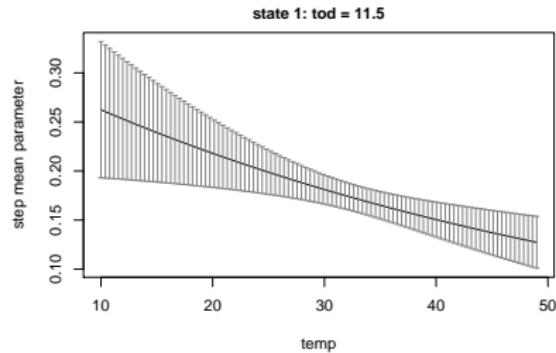
```
plot(m3, plotCI=TRUE)
```

Transition probabilities: tod = 11.5

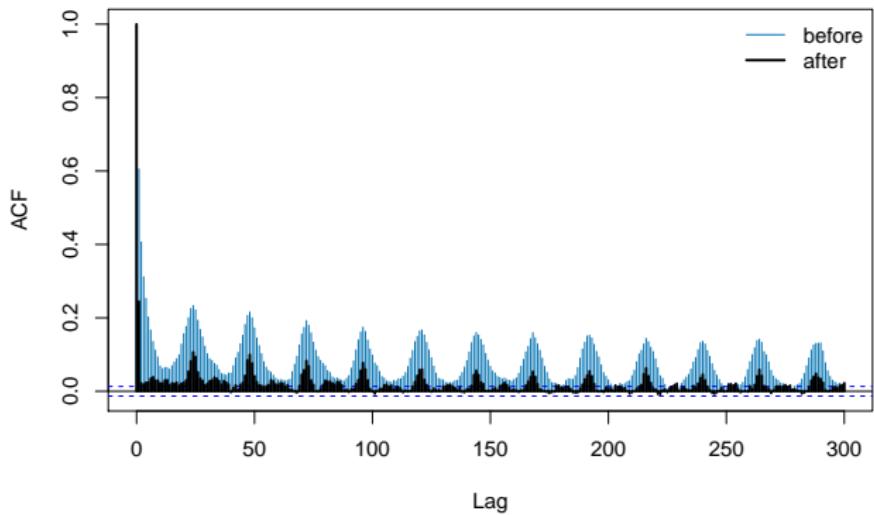


Covariates: temperature and time of day

```
plot(m3, plotCI=TRUE)
```



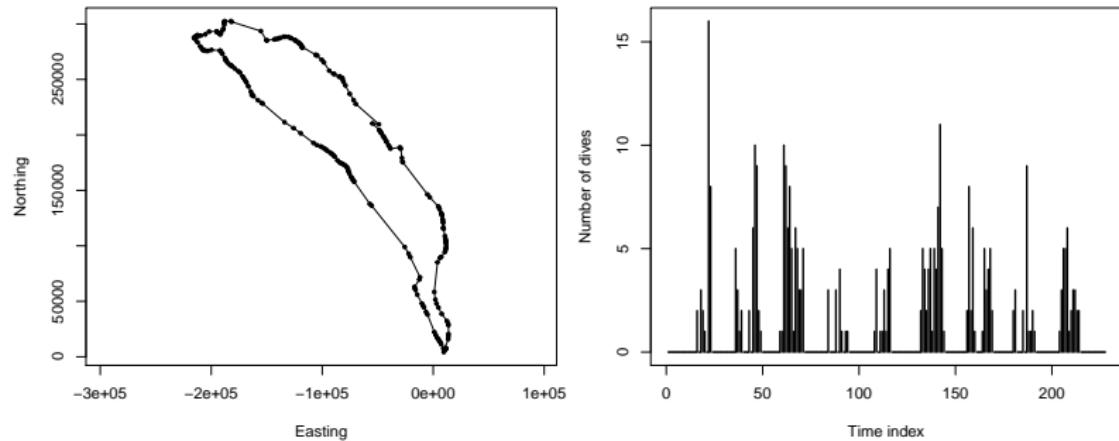
Step length residuals



1. Elephant example
2. Northern fur seal example
3. Other examples

Northern fur seal data

- ▶ Locations (irregular sampling frequency)
- ▶ Number of “foraging” 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:

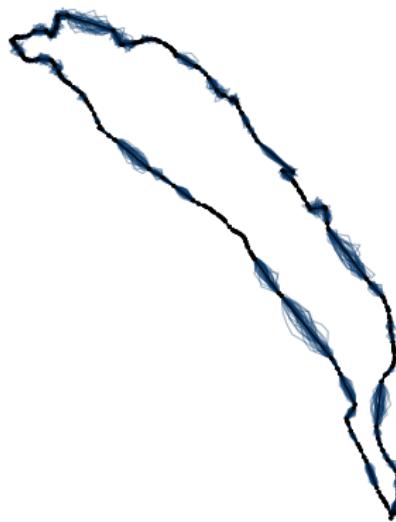
- ▶ Fit continuous-time movement model (`crawl`)
- ▶ Draw temporally-regular realisations from the fitted model
- ▶ Fit a HMM to each realisation



Multiple imputation

Solution to irregular sampling and measurement error:

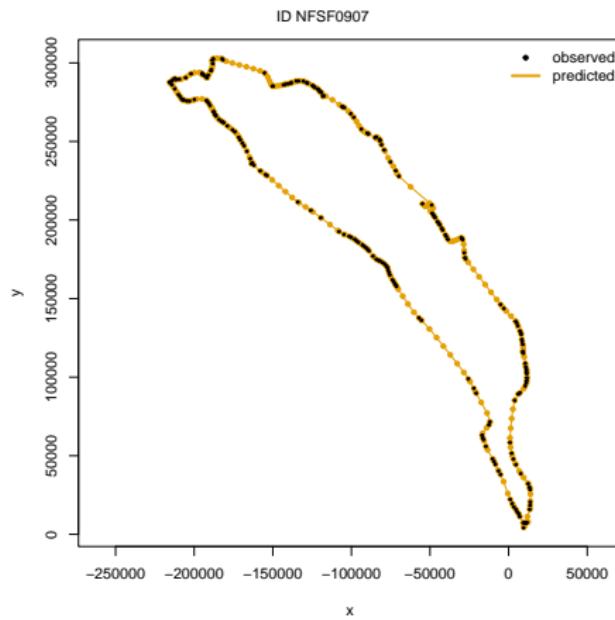
- ▶ Fit continuous-time movement model (`crawl`)
- ▶ Draw temporally-regular realisations from the fitted model
- ▶ Fit a HMM to each realisation



Fit crawl model

```
crwOut <- crawlWrap(obsData=nfsData, predTime="1 hour")
```

```
plot(crwOut)
```



Multiple imputation multivariate HMM

Step length: $L_t | S_t = s \sim \text{Gamma}(\mu_s, \sigma_s)$

Turn angle: $\phi_t | S_t = s \sim \text{wCauchy}(0, \rho_s)$

of dives: $\delta_t | S_t = s \sim \text{Poisson}(\lambda_s)$

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

# initial parameters
stepPar0 <- c(500, 1000, 5000, 1000, 1000, 2000)
anglePar0 <- c(0.01, 0.05, 0.75)
divePar0 <- c(10e-4, 2, 10e-4)
Par0 <- list(step=stepPar0, angle=anglePar0, dive=divePar0)
```

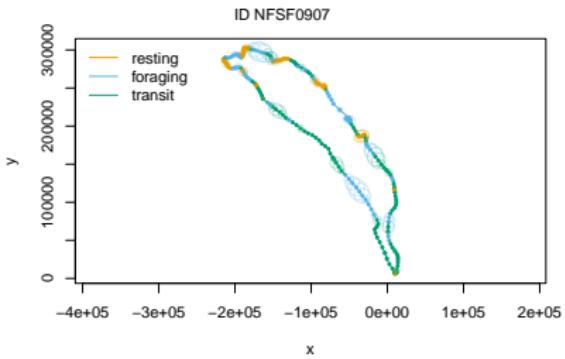
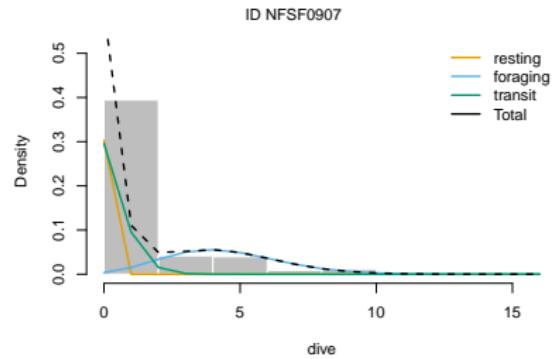
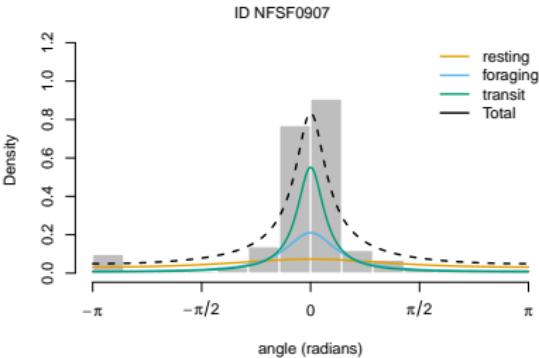
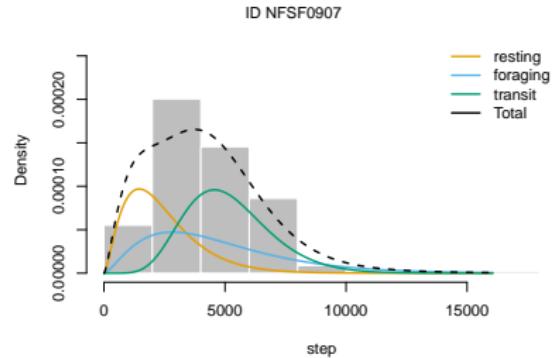
Multiple imputation multivariate HMM

```
# fit 3-state model
m_nfs <- MIfitHMM(crwOut, nSims = 30,
                     nbStates = 3, dist = dist, Par0 = Par0)
```

```
## Drawing 30 realizations from the position process using crawl...
## Fitting 30 realizations of the position process using fitHMM...
## Fitting 30 imputations in parallel...
## Decoding state sequences for each imputation...
## Decoding state probabilities for each imputation...
## Calculating location 95% error ellipses... DONE
```

Results – NFS model

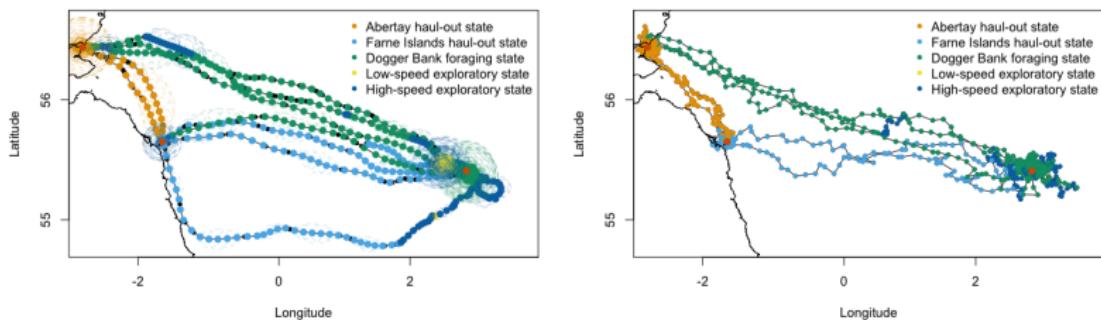
plot(m_nfs)



1. Elephant example
2. Northern fur seal example
3. Other examples

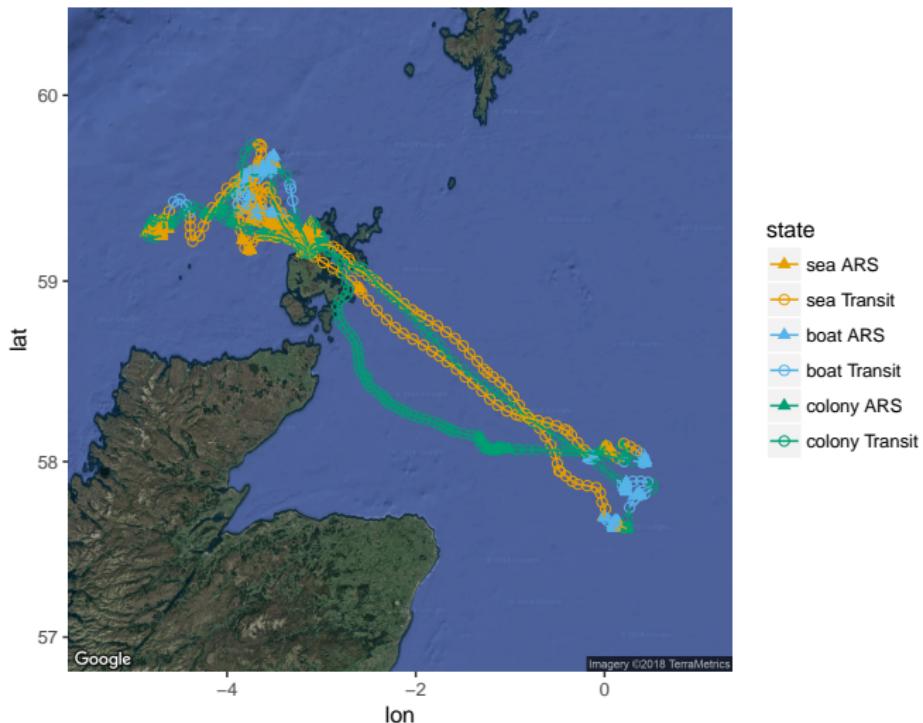
Biased correlated random walks

- ▶ Dynamic or static centres of attraction
- ▶ Simulation from a fitted model



Data from: McClintock et al. (2012), "A general discrete-time modeling framework for animal movement using multistate random walks", Ecological Monographs.

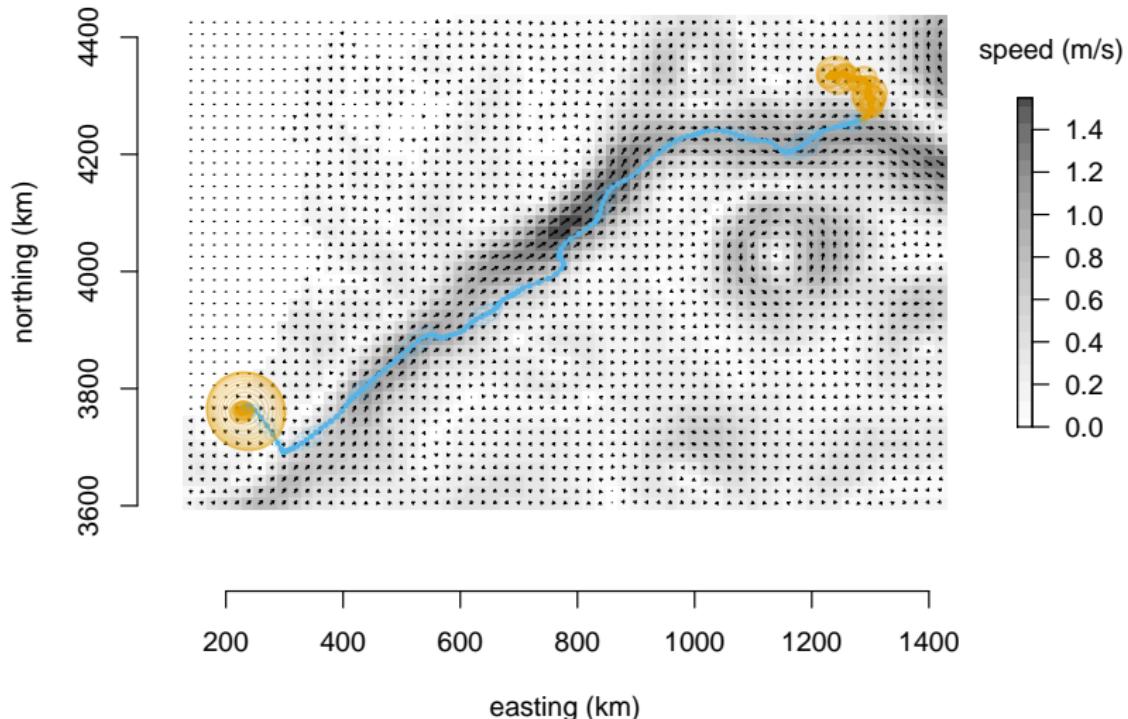
Dynamic activity centres



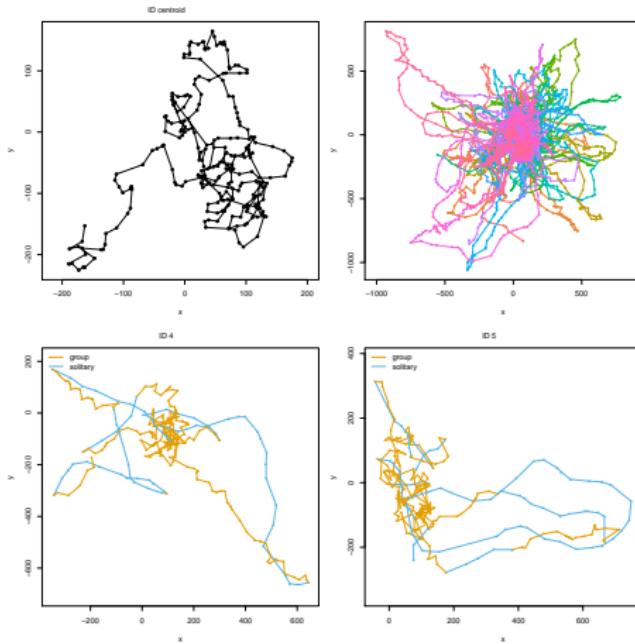
Data from: Pirotta et al. (2018), “Central place foragers and moving stimuli: A hidden-state model to discriminate the processes affecting movement”, *J Anim Ecol.*

Spatio-temporal covariate raster data

- ▶ e.g. ocean currents, wind velocity, sea ice concentration

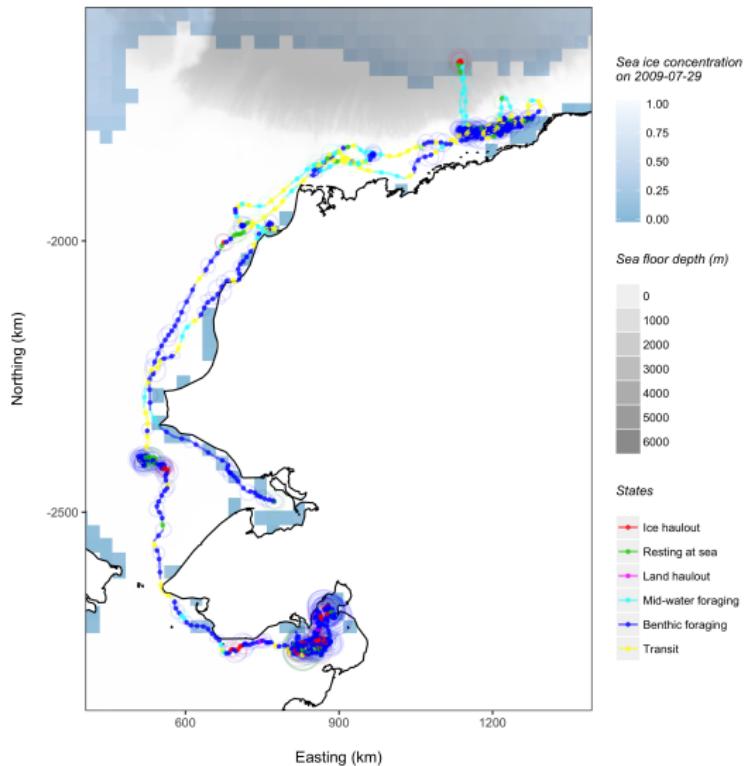


Group dynamic models



Langrock et al. (2014), “Modelling group dynamic animal movement”, *Methods Ecol Evol.*

Auxiliary biotelemetry and environmental data



Data from: McClintock et al. (2017). "Bridging the gaps in animal movement: hidden behaviors and ecological relationships revealed by integrated data streams", *Ecosphere*, 8(3).

Questions?

