HMM Workshop

Rocío Joo

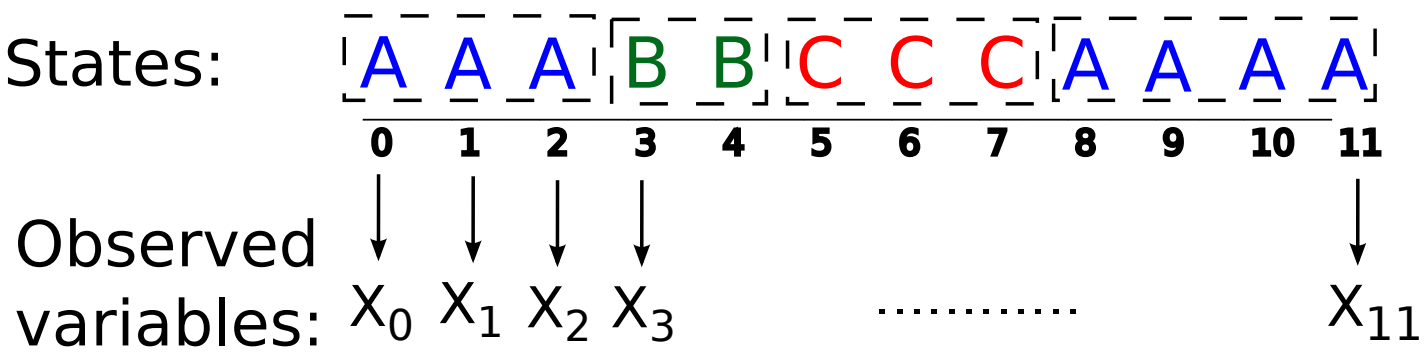
Structure of the workshop

- Introduction to HMMs
- Data for application
- 3 HMM applications
  - Simple HMM with 3 states
  - HMM with 4 states with constraints in transitions
  - HMM with covariates conditioning transitions and observed variables
- Other stuff you can and can't do with momentuHMM
- References

Hidden Markov Models:

- What are they?
- What assumptions?

Hidden Markov Models:



Hidden Markov Models:

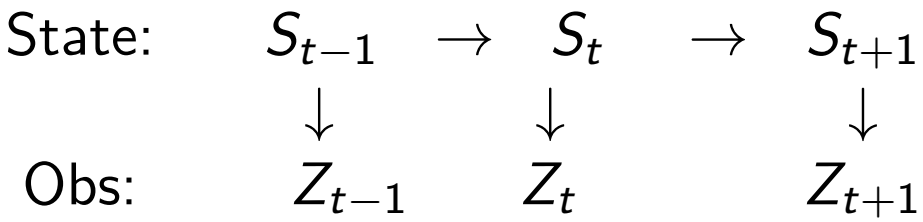
$$
\begin{array}{lccccc}
\text{State:} & S_{t-1} & \rightarrow & S_t & \rightarrow & S_{t+1} \\
 & \downarrow & & \downarrow & & \downarrow \\
\text{Obs:} & Z_{t-1} & & Z_t & & Z_{t+1}
\end{array}
$$

Hidden Markov Models:

$$\begin{array}{lccccc} \text{State:} & S_{t-1} & \rightarrow & S_t & \rightarrow & S_{t+1} \\ & \downarrow & & \downarrow & & \downarrow \\ \text{Obs:} & Z_{t-1} & & Z_t & & Z_{t+1} \end{array}$$

- Underlying unobserved Markov process $\{S_t\}$:
  $P(S_t = s_t \mid S_{t-1} = s_{t-1}) = P(S_t = j \mid S_{t-1} = i) = \gamma_{ij}$

Hidden Markov Models:



- Underlying unobserved Markov process $\{S_t\}$:
  $P(S_t = s_t \mid S_{t-1} = s_{t-1}) = P(S_t = j \mid S_{t-1} = i) = \gamma_{ij}$

- State-dependent observation process $\{Z_t\}$: $P(Z_t = z_t \mid S_t = s_t)$

Data for application: seals

- Southern elephant seal (*Mirounga leonina*)
- Trips from 15 animals (eight adult females and seven subadult males) tagged with Argos at Kerguelen Island.
- Previous filtering with SSMs was performed.

Let's start preparing the data for modelling first. We need observed variables.

```
path_data <- "./Data/"
trackData <- read.csv(paste0(path_data,"seal_tracks.csv"))
```

```r
head(trackData)
```

```
##   ID      lon       lat                 date
## 1  1 70.60946 -49.60737 2014-02-11 10:34:58
## 2  1 70.82908 -50.08287 2014-02-11 20:10:58
## 3  1 70.90029 -50.32835 2014-02-12 05:46:58
## 4  1 70.85766 -50.54746 2014-02-12 15:22:58
## 5  1 70.63792 -50.90529 2014-02-13 00:58:58
## 6  1 70.48480 -50.99666 2014-02-13 10:34:58
```

```r
str(trackData)
```

```
## 'data.frame':    7353 obs. of  4 variables:
##  $ ID  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ lon : num  70.6 70.8 70.9 70.9 70.6 ...
##  $ lat : num  -49.6 -50.1 -50.3 -50.5 -50.9 ...
##  $ date: Factor w/ 7353 levels "2013-02-13 22:27:43",..: 44
```

Just checking if the time steps are regular

```
dates <- as.character(trackData[,4])
time_dif <- diff.Date(as.POSIXct(dates, tz = "GMT"))
head(time_dif)
```

```
## Time differences in hours
## [1] 9.6 9.6 9.6 9.6 9.6 9.6
```

Calling the package

```
library(momentuHMM)
```

Now, use the prepData for two reasons:

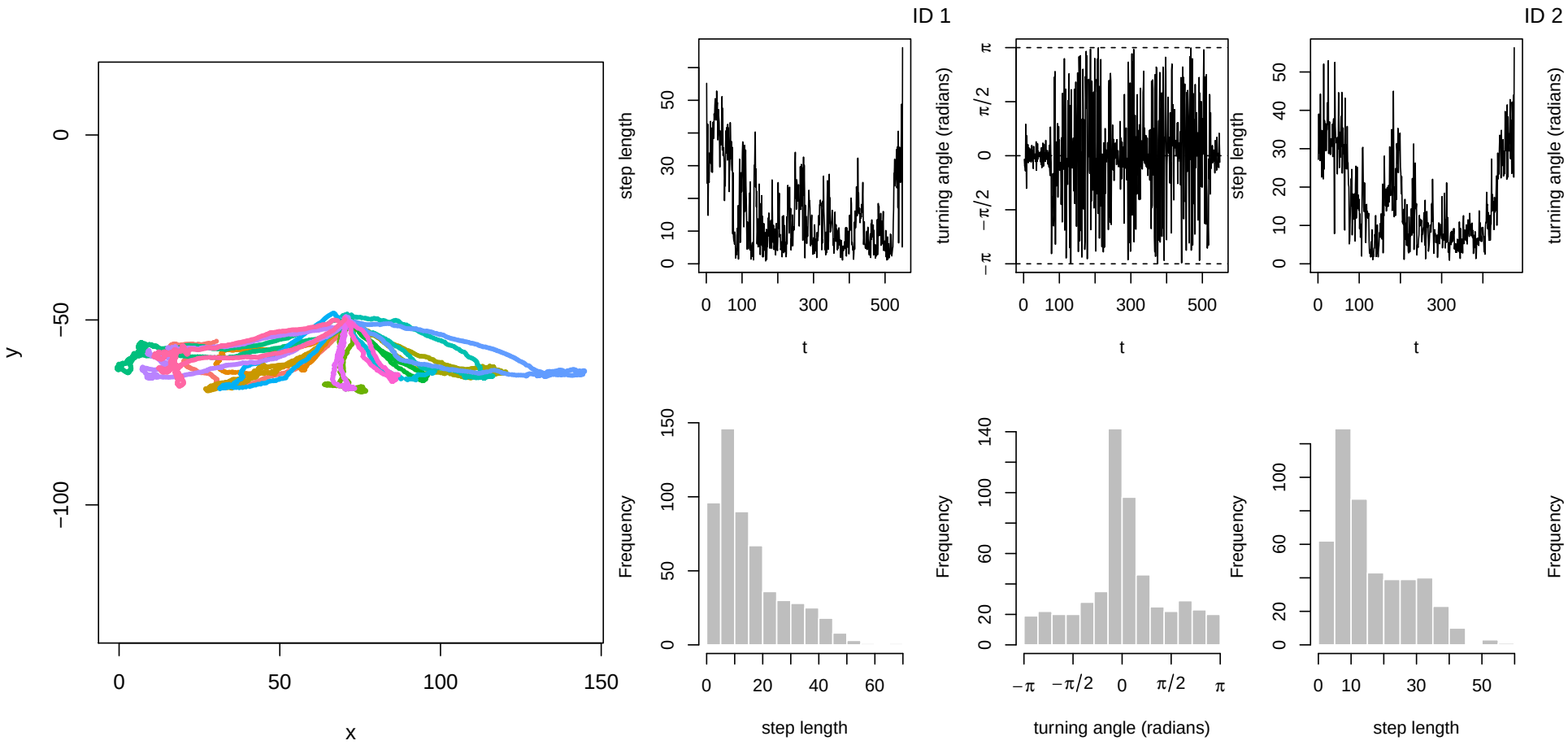
1. derive the observed variables through that function
2. transform the dataframe into a momentuHMMData object

```r
data <- prepData(trackData, type = "LL",
                 coordNames = c("lon", "lat"))
head(data)
```

```
##   ID     step      angle                date        x
## 1  1 55.19656         NA 2014-02-11 10:34:58 70.60946 -49.6
## 2  1 27.77600 -0.1051839 2014-02-11 20:10:58 70.82908 -50.0
## 3  1 24.56027 -0.3069499 2014-02-12 05:46:58 70.90029 -50.3
## 4  1 42.72315 -0.2468352 2014-02-12 15:22:58 70.85766 -50.5
## 5  1 14.80219 -0.4426067 2014-02-13 00:58:58 70.63792 -50.9
## 6  1 27.87312  0.9141237 2014-02-13 10:34:58 70.48480 -50.9
```

You can also plot the tracks and the series of steps and angles with

```
plot(x=data,compact=TRUE,ask=FALSE)
```



Modeling movement with a 3-state HMM

- 3 movement modes are expected: travelling (T), searching (S) and foraging (F)
- Observation variables: step length and turning angles
- No covariates

Parametrisation of the model:

Transition probability matrix:

$$\Gamma = \begin{bmatrix} \gamma_{TT} & \gamma_{TS} & \gamma_{TF} \\ \gamma_{ST} & \gamma_{SS} & \gamma_{SF} \\ \gamma_{FT} & \gamma_{FS} & \gamma_{FF} \end{bmatrix}$$

6 parameters to estimate

Modeling movement with a 3-state HMM

Parametrisation of the model:

L: step length

$L_t|S_t = s_t \sim Gamma(\text{shape}, \text{scale})$

initial values:

$L_t|S_t = T \sim Gamma(40, 10)$

$L_t|S_t = S \sim Gamma(20, 8)$

$L_t|S_t = F \sim Gamma(8, 5)$

Modeling movement with a 3-state HMM

Parametrisation of the model:

Θ: turning angle

$\Theta_t | S_t = s_t \sim VM(\text{location}, \text{concentration})$

initial values:
$\Theta_t | S_t = T \sim VM(-0.001, 7)$
$\Theta_t | S_t = S \sim VM(0, 4)$
$\Theta_t | S_t = F \sim VM(\pi, 0.1)$

So 12 parameters to estimate for the observation model. 18 in total.

Modeling movement with a 3-state HMM

Now in R!

Initial values for parameters in the model

```
stateNames <- c("travel","search","forage")
shape_0 <- c(40,20,8)
scale_0 <- c(10,8,5)
stepPar0 <- c(shape_0,scale_0)
location_0 <- c(-0.001,0,3.14)
concentration_0 <- c(7,4,0.1)
anglePar0 <- c(location_0,concentration_0)
```

Modeling movement with a 3-state HMM

And fitting the model

```
m1 <- fitHMM(data=data, nbStates=3,
             dist=list(step="gamma",angle="vm"),
             Par0=list(step=stepPar0, angle=anglePar0),
             estAngleMean = list(angle=TRUE),
             stateNames = stateNames)
print(m1) #estimations
```

Modeling movement with a 3-state HMM

Reconstructing state sequences:

The Viterbi algorithm:

$$arg\ max_{S_0=s_0,\ldots,S_t=s_t} P(S_0 = s_0, \ldots, S_t = s_t \mid Z_0 = z_0, \ldots, Z_t = z_t)$$

In R:

```
states.1 <- as.factor(viterbi(m1))
prop.table(table(states.1))
```

```
## states.1
##         1         2         3
## 0.2623419 0.2560860 0.4815721
```

Modeling movement with a 3-state HMM

You might want to compute the probability of the process being in a different state at each time point...

```
stateProbs(m1)
```

It is computing $P(S_t = s_t \mid Z_0 = z_0, ..., Z_t = z_t, ..., Z_{T-1} = z_{T-1})$

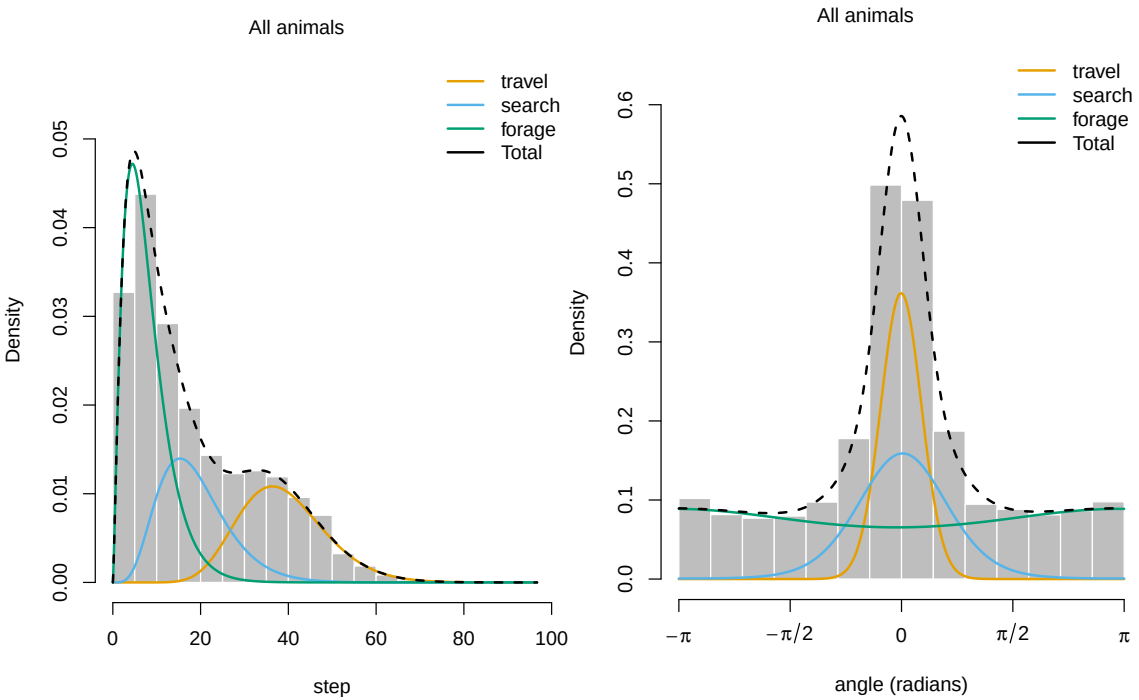
which is different than what viterbi does.

Modeling movement with a 3-state HMM

Plotting outputs

```
plot(m1, plotCI = TRUE, plotTracks=FALSE)
```

```
## Decoding state sequence... DONE
```

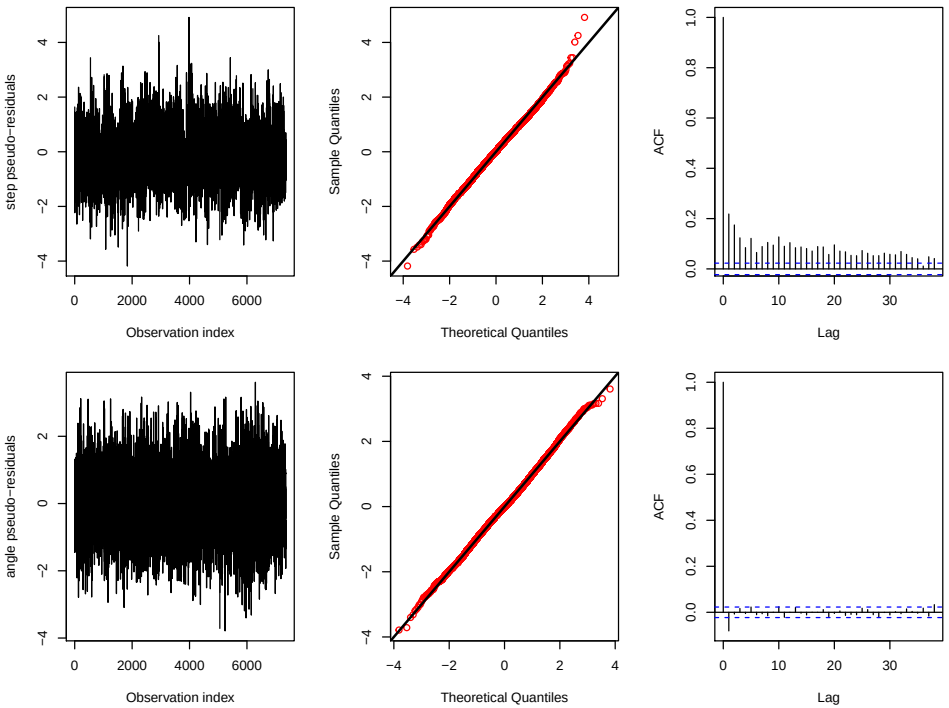


Modeling movement with a 3-state HMM

Checking residuals

```
plotPR(m1)
```

```
## Computing pseudo-residuals... DONE
```



Modeling movement with a 4-state HMM

- 4 movement modes are expected: outbound (O), searching (S), foraging (F) and inbound (I)
- Not all transitions are possible
- Observation variables: step length and turning angles
- No covariates

The observation model is almost the same, we add one state

Modeling movement with a 4-state HMM

Parametrisation of the model:

L: step length

$L_t|S_t = s_t \sim Gamma(\text{shape}, \text{scale})$

initial values:

$L_t|S_t = O \sim Gamma(40, 10)$
$L_t|S_t = S \sim Gamma(20, 8)$
$L_t|S_t = F \sim Gamma(8, 5)$
$L_t|S_t = I \sim Gamma(40, 10)$

Modeling movement with a 4-state HMM

Parametrisation of the model:

Θ: turning angle

$\Theta_t | S_t = s_t \sim VM(\text{location}, \text{concentration})$

initial values:
$\Theta_t | S_t = O \sim VM(-0.001, 7)$
$\Theta_t | S_t = S \sim VM(0, 4)$
$\Theta_t | S_t = F \sim VM(\pi, 0.1)$
$\Theta_t | S_t = I \sim VM(-0.001, 7)$

So 16 parameters to estimate for the observation model.

Modeling movement with a 4-state HMM

In R:

```
stateNames <- c("outbound","search","forage","inbound")
shape_0 <- c(40,20,8,40)
scale_0 <- c(10,8,5,10)
stepPar0 <- c(shape_0,scale_0)
location_0 <- c(-0.001,0,3.14,-0.001)
concentration_0 <- c(7,4,0.1,7)
anglePar0 <- c(location_0,concentration_0)
```

Modeling movement with a 4-state HMM

Parametrisation of the model:

Not all transitions are possible:

	O	S	F	I
O			0	0
S	0			
F	0			0
I	0	0	0	

Modeling movement with a 4-state HMM

Parametrisation of the model:

Not all transitions are possible:

	O	S	F	I
O	γ_{OO}	γ_{OS}	0	0
S	0	γ_{SS}	γ_{SF}	γ_{SI}
F	0	γ_{FS}	γ_{FF}	0
I	0	0	0	1

4 parameters to estimate. 20 in total.

Modeling movement with a 4-state HMM

Parametrisation of the model:

Not all transitions are possible:

	O	S	F	I
O	γ_{OO}	γ_{OS}	0	0
S	0	γ_{SS}	γ_{SF}	γ_{SI}
F	0	γ_{FS}	γ_{FF}	0
I	0	0	0	1

4 parameters to estimate. 20 in total.

In momentuHMM, $\gamma_{ij} = logit^{-1}(\beta X)$; X: covariates

If there are no covariates, then $\gamma_{ij} = logit^{-1}(\beta_0) = \frac{1}{1 + \exp -\beta_0}$

Then, for $\gamma_{ij} \approx 0$, β_0: large negative number; e.g. -100

Thus, for our case:

```
beta_M2=c(NA,-100,-100, -100, NA, NA, -100, NA, -100, -100,
        -100, -100)
```

Modeling movement with a 4-state HMM

And fitting the model

```
m2 <- fitHMM(data=data, nbStates=4,
             dist=list(step="gamma",angle="vm"),
             Par0=list(step=stepPar0, angle=anglePar0),
             estAngleMean = list(angle=TRUE),
             fixPar=list(beta=beta_M2),
             stateNames = stateNames)
print(m2) #estimations
```

Modeling movement with a 4-state HMM

Reconstructing state sequences:

The Viterbi algorithm:

```
states.2 <- as.factor(viterbi(m2))
prop.table(table(states.2))
```

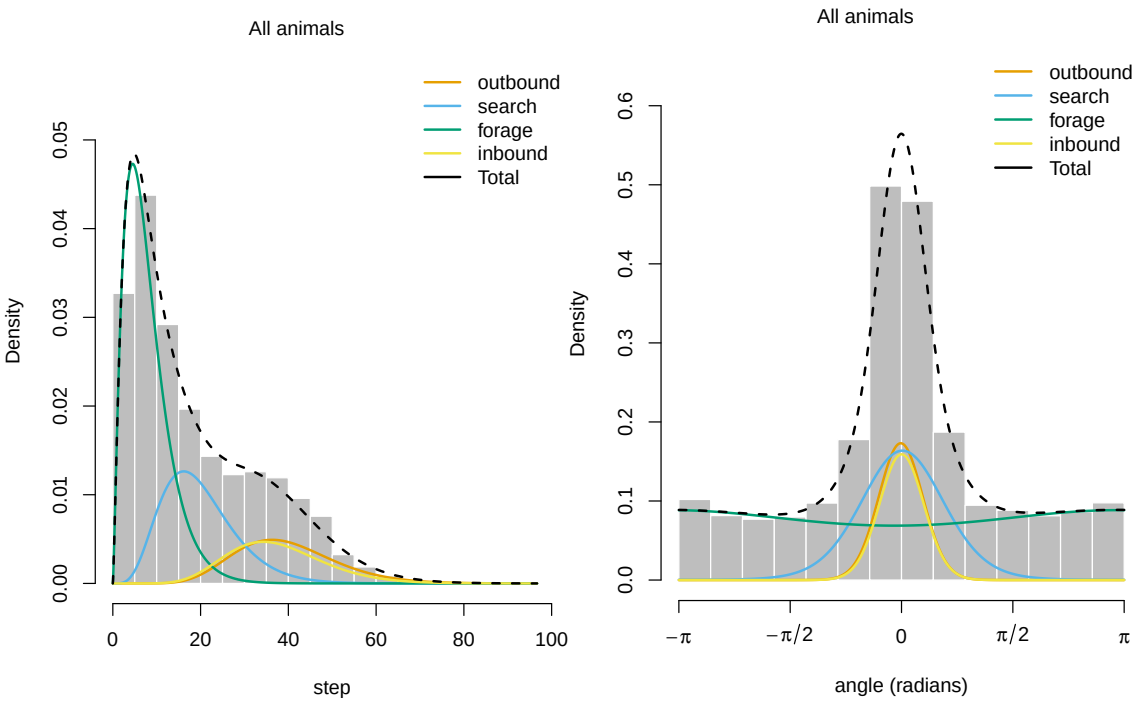
```
## states.2
##         1         2         3         4
## 0.1365429 0.2461580 0.4920441 0.1252550
```

Modeling movement with a 4-state HMM

Plotting outputs

```
plot(m2, plotCI = TRUE, plotTracks=FALSE)
```

```
## Decoding state sequence... DONE
```

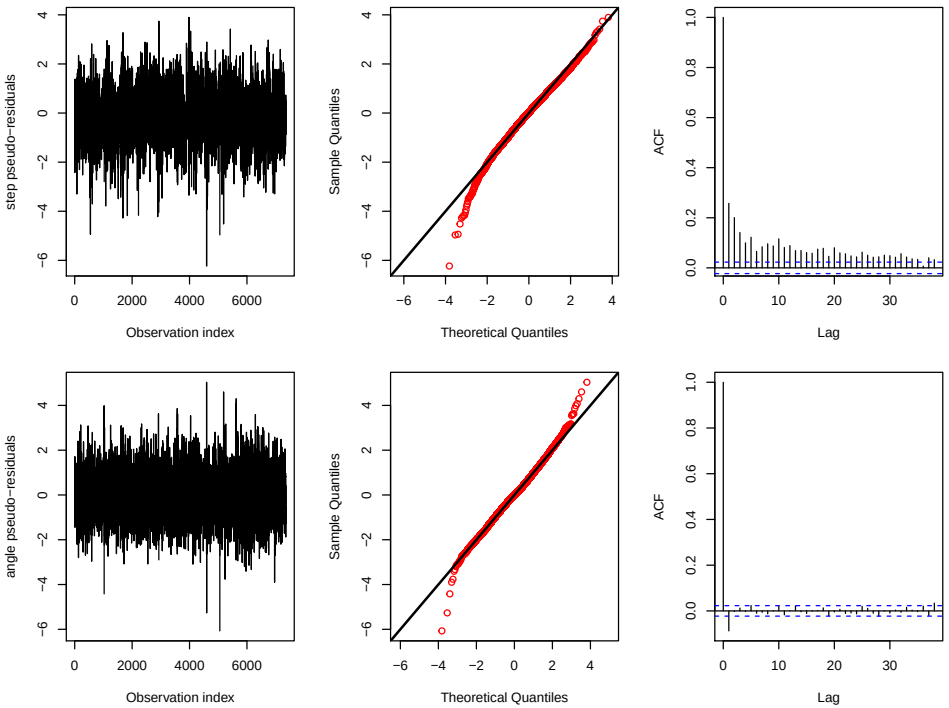


Modeling movement with a 4-state HMM

Checking residuals

```
plotPR(m2)
```

```
## Computing pseudo-residuals... DONE
```



4-state HMM with covariates

- 4 movement modes are expected: outbound (O), searching (S), foraging (F) and inbound (I)
- Not all transitions are possible
- Observation variables: step length and turning angles
- What if transitioning from outbound or into inbound depended on covariates?
- What if the observed model depended on covariates as well?
- Covariates: distance to the colony, time since departure from the colony

Let's start by deriving the covariates in R:

4-state HMM with covariates

We first to compute distances to the colony:

```
center <- matrix(c(70,-49),nrow=1,dimnames=list("colony"))
data <- prepData(data=trackData, type="LL", centers=center,
                 coordNames = c("lon", "lat"))
head(data[,c(1:3,7,8)])
```

```
##   ID     step       angle colony.dist colony.angle
## 1  1 55.19656          NA    80.79235    0.5847407
## 2  1 27.77600 -0.1051839   134.55555   -2.9626223
## 3  1 24.56027 -0.3069499   161.40565   -2.9047384
## 4  1 42.72315 -0.2468352   182.86506   -2.6678926
## 5  1 14.80219 -0.4426067   216.80945   -2.5543702
## 6  1 27.87312  0.9141237   224.79045   -2.1702936
```

4-state HMM with covariates

Deriving time spent since it left the colony:

```
time <- NULL
for(id in unique(data$ID)) {
nbSubObs <- sum(data$ID==id)
# approximately in months for interval = 9.6h
time <- c(time, (1:nbSubObs)*9.6/(24*30))
}
data$time <- time
```

Now, back to the parameters of the model.

4-state HMM with covariates

Parametrisation of the model:

Not all transitions are possible:

	O	S	F	I
O	γ_{OO}	γ_{OS}	0	0
S	0	γ_{SS}	γ_{SF}	γ_{SI}
F	0	γ_{FS}	γ_{FF}	0
I	0	0	0	1

4-state HMM with covariates

Parametrisation of the model:

Not all transitions are possible:

	O	S	F	I
O	γ_{OO}	γ_{OS}	0	0
S	0	γ_{SS}	γ_{SF}	γ_{SI}
F	0	γ_{FS}	γ_{FF}	0
I	0	0	0	1

Michelot et al (2017): '(...) Animals often make fast trips away from the colony and tend to switch into other movement modes once they have reached foraging grounds'

4-state HMM with covariates

Parametrisation of the model:

Not all transitions are possible:

	O	S	F	I
O	γ_{OO}	γ_{OS}	0	0
S	0	γ_{SS}	γ_{SF}	γ_{SI}
F	0	γ_{FS}	γ_{FF}	0
I	0	0	0	1

Michelot et al (2017): '(...) Animals often make fast trips away from the colony and tend to switch into other movement modes once they have reached foraging grounds'

$$\gamma_{OS}^{(t)} = logit^{-1}(\beta_0^{(OS)} + \beta_1^{(OS)} d_t)$$

d_t: distance of the animal to the colony at time t

4-state HMM with covariates

Parametrisation of the model:

Not all transitions are possible:

	O	S	F	I
O	γ_{OO}	γ_{OS}	0	0
S	0	γ_{SS}	γ_{SF}	γ_{SI}
F	0	γ_{FS}	γ_{FF}	0
I	0	0	0	1

Michelot et al (2017): '(...) Animals often make fast trips away from the colony and tend to switch into other movement modes once they have reached foraging grounds'

$$\gamma_{OS}^{(t)} = logit^{-1}(\beta_0^{(OS)} + \beta_1^{(OS)} d_t)$$

d_t: distance of the animal to the colony at time t

$$\gamma_{OS}^{(t)} = \frac{\exp(\beta_0^{(OS)} + \beta_1^{(OS)} d_t)}{1 + \exp(\beta_0^{(OS)} + \beta_1^{(OS)} d_t)}$$

4-state HMM with covariates

Parametrisation of the model:

Not all transitions are possible:

	O	S	F	I
O	γ_{OO}	γ_{OS}	0	0
S	0	γ_{SS}	γ_{SF}	γ_{SI}
F	0	γ_{FS}	γ_{FF}	0
I	0	0	0	1

Michelot et al (2017): '... time constraints to go back to the colony ...'

4-state HMM with covariates

Parametrisation of the model:

Not all transitions are possible:

	O	S	F	I
O	γ_{OO}	γ_{OS}	0	0
S	0	γ_{SS}	γ_{SF}	γ_{SI}
F	0	γ_{FS}	γ_{FF}	0
I	0	0	0	1

Michelot et al (2017): '... time constraints to go back to the colony ...'

$$\gamma_{SI}^{(t)} = \frac{\exp\left(\beta_0^{(SI)} + \beta_1^{(SI)}(t - t_0)\right)}{1 + \exp\left(\beta_0^{(SI)} + \beta_1^{(SI)}(t - t_0)\right) + \exp\left(\beta_0^{SF}\right)}$$

$$\gamma_{SF}^{(t)} = \frac{\exp\left(\beta_0^{(SF)}\right)}{1 + \exp\left(\beta_0^{(SI)} + \beta_1^{(SI)}(t - t_0)\right) + \exp\left(\beta_0^{SF}\right)}$$

4-state HMM with covariates

Parametrisation of the model:

What β parameters to estimate? NA to those we need, 0 to those we don't, and -100 when transitions are not likely to occur

	OS	OF	OI	SO	SF	SI	FO	FS	FI	IO	IS	IF
β_0												
d_t												
$t - t_0$												

4-state HMM with covariates

Parametrisation of the model:

What β parameters to estimate? NA to those we need, 0 to those we don't, and -100 when transitions are not likely to occur

In R:

```
beta <- matrix(c(NA,-100,-100,-100,NA,NA,-100,NA,-100,
                 -100,-100,-100,
                 NA,0,0,0,0,0,0,0,0,0,0,0,
                 0,0,0,0,0,NA,0,0,0,0,0,0),
               nrow=3,byrow=TRUE)
formula <- ~ colony.dist + time
```

4-state HMM with covariates

Parametrisation of the model:

What about the observed variables?

Idea: model outbound and inbound as BRWs, with repulsion from the colony for outbound and attraction towards the colony during inbound.

In momentuHMM, you can express it by:

```
angleFormula <- ~ state1(colony.angle) +
  state4(colony.angle)
fixPar <- list(angle=c(-100,100,NA,NA,NA,NA),beta=beta)
```

-100: mean direction away from the colony

100: mean direction towards the colony

4 concentration parameters to estimate

4-state HMM with covariates

Parametrisation of the model:

Gamma distributions will be used for step length.

Instead of starting all over, we use the estimated values by the previous model as initial values to fit this model

```
m2 <- fitHMM(data=data, nbStates=4,
             dist=list(step="gamma",angle="vm"),
             Par0=list(step=stepPar0, angle=anglePar0),
             estAngleMean = list(angle=TRUE),
             fixPar=list(beta=beta_M2),
             stateNames = stateNames)
Par0 <- getPar0(model=m2, nbStates=4,
                DM=list(angle=list(mean=angleFormula,
                                   concentration=~1)),
                estAngleMean=list(angle=TRUE),
                circularAngleMean=list(angle=TRUE),
                formula=formula)
```

4-state HMM with covariates

Now, fitting the model:

```
m3 <- fitHMM(data=data, nbStates=4,
             dist=list(step="gamma",angle="vm"),
             Par0=list(step=Par0$Par$step,
                       angle=Par0$Par$angle),
             beta0=Par0$beta, fixPar=fixPar, formula=formula,
             DM=list(angle=list(mean=angleFormula,
                                concentration=~1)),
             estAngleMean=list(angle=TRUE),
             circularAngleMean=list(angle=TRUE),
             stateNames = stateNames)
```

4-state HMM with covariates

Reconstructing state sequences:

The Viterbi algorithm:

```
states.3 <- as.factor(viterbi(m3))
prop.table(table(states.3))
```

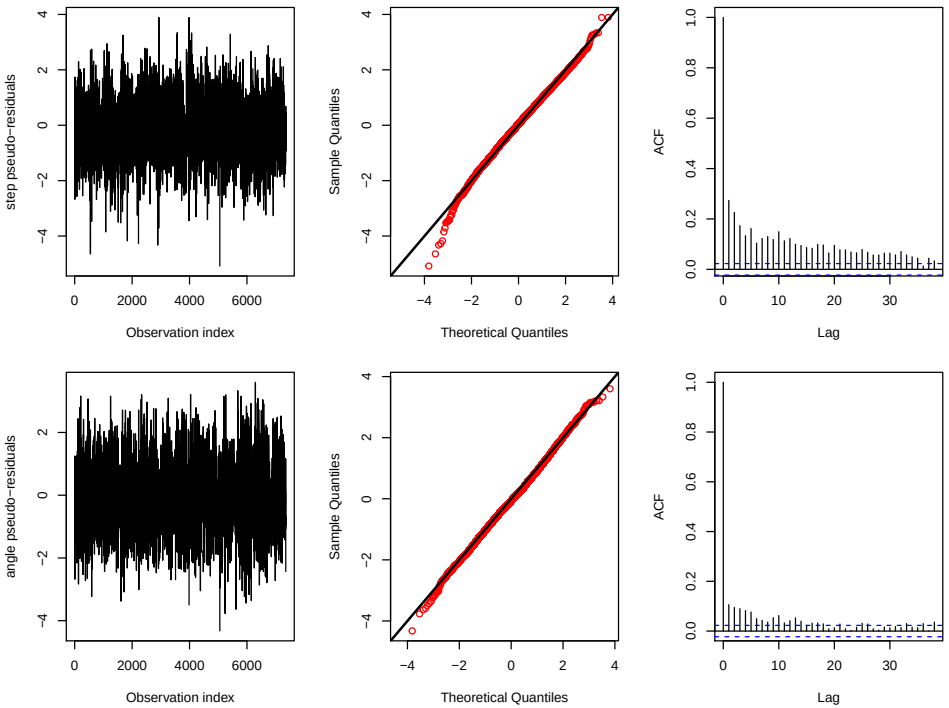
```
## states.3
##         1         2         3         4
## 0.1262070 0.2437101 0.5154359 0.1146471
```

4-state HMM with covariates

Checking residuals

```
plotPR(m3)
```

```
## Computing pseudo-residuals... DONE
```



Other stuff that you can do

- Introduce a dynamic activity center
- More complicated models (e.g. parameter bounds)
- Data imputation methods (e.g. using crawl)

Other stuff that you can do

- Introduce a dynamic activity center
- More complicated models (e.g. parameter bounds)
- Data imputation methods (e.g. using crawl)

Stuff that cannot be done in momentuHMM

- Hidden semi-Markov models
- Random effects

Acknowledgments:

Théo Michelot

Thanks!