



Alan Eduardo de Barros

(De Barros, Alan E.)

Aluno de Doutorado pela Ecologia IB-USP

Orientador: Paulo Inácio, **Co-orientador:** Ronaldo Morato

Titulo da Tese:

Individual-Based Movement Models and Integrated Step Selection Functions applied to Biodiversity Conservation: The jaguar (*Panthera onca*) as a case study

In my thesis we will discuss the importance of including animal movements within Conservation Planning and propose a general framework which could be applicable to any animal species and location of interest. In Chapter 1 we will develop such framework and develop Individual Based Movement Models which will incorporate probabilistic changes in movement behaviours. This will be done controlling key parameters (i.e. movement distance, directional persistence and habitat preference) to generate biased random walks. We will explore changes in the parameters and the consequences of it on resultant movement trajectories. In the following chapters we will refine our model and apply it to jaguars' movements. First, in Chapter 2 we will use the jaguar dataset to explore and estimate movement metrics and resource selection at multiple scales. Then, in Chapter 3 we will use our framework to build jaguars' IBMMs based on these metrics and selected landscape variables. Furthermore, understanding how movements could contribute to refine corridor proposals. Finally, on Chapter 4 we will simulate changes on the landscape looking for critical thresholds on variables such as forest cover and how changes on these variables interfere on jaguars' movements and ultimately on their survival.

Meus exercícios

[exec](#)

Trabalho final

Plano A: "MoveMovie" Criando gifs de movimentos

Contextualização

Movimentos são um comportamento fundamental para sobrevivência da maioria das espécies, pois

Ihles permite procurar e encontrar os melhores locais para sobrevivência e reprodução (Grimm & Railsback 2005). A Ecologia de Movimentos é uma disciplina ainda em expansão e o surgimento e emprego de tecnologias como colares GPS tem sido um divisor de águas nesse campo, pois permite a aquisição de várias localizações a intervalos regulares. E com a abundância de localizações, novos métodos têm sido propostos, desde de novas análises de área de vida, considerando e permitindo a auto-correlação dos dados (pacote ctm), assim como novas metodologias relacionadas a análises de trajetórias e de seleção de recursos. Apesar de haver vários pacotes do R relacionados a diferentes tipos de análises de movimento (adehabitatLT, move), incluindo possibilidades de outputs gráficos com os dados e análises plotados espacialmente, não existe uma solução pronta, por exemplo, para plotar os dados de maneira sequencial 1 a 1, chamando ao mesmo tempo um mapa geográfico (do google) (maptools), ggplot2) e permitindo salvar a movimentação sequencialmente na forma de vídeo ou gif (animation). No entanto, reunindo recursos de diferentes pacotes e criando alguns comandos para estabelecer a ligação entre eles isso pode ser feito.

Objetivo: Criar uma função capaz de **1)** plotar as localizações de forma sequencial, **2)** permitindo visualizar a movimentação de animais na paisagem, **3)** chamando um mapa de fundo através do google maps ou (ou google earth) e salvando a animação em gif (ou vídeos).

Nome da função: MoveMovie

Dados de entrada: **a)** dataframe com dados de movimento, **b)** fotografia aérea como mapa (google maps ou google earth) baseada nas dimensões espaciais de **a)**.

Principais parâmetros de entrada (dados filtrados e renomeados preliminarmente):

a) Dataframe

- Item de lista não ordenada `id` = `individual.local.identifier`, corresponde ao número de identificação única do indivíduo
- `timestamp` = define a sequência temporal, com Mês/Dia/Ano horas e minutos
- `long` = `x`, posição espacial no eixo `x` com base na longitude
- `lat` = `y`, posição espacial no eixo `y` com base na latitude
- Demais parâmetros de interesse podem ser mantidos ou excluídos, não interferirão na função

b) plotagem de imagem com dimensões e projeção especificadas por a)

Pseudo-código:

1. Item de lista ordenada `id` de lista ordenada `id` Cria objeto1 dando nome ao dataframe.
2. Item de lista ordenada `id` Instala todos os packages necessários.
3. Ajusta/ formata dataframe para ser entendido e lido como informação espacial (uma possibilidade seria utilizando o package `move`)
4. E cria objeto2 correspondente `objeto2=move(x=, y=,time=as.POSIXct( $timestamp,format="%Y-%m-%d %H:%M:%S",tz="UTC"),data=id, proj=CRS("+proj=longlat"),animal= $id)`
5. Plota localizações
6. Define área de imagem como base e utiliza algum package capaz de baixar imagem aérea do google maps
7. Define objeto da imagem
8. Plota imagem
9. Ajusta/ formata objeto2 para produzir animação (utilizando packages específicos) plotando uma

localização por vez.

10. Utiliza packages específicos para salvar vídeo ou gif do movimento do animal¹ na pasta especificada
11. Cria um ciclo para repetir o processo com arquivos de outros animais e salvando todos os gifs.

Saída:

```
Parcial 1 Plotagem dos pontos de localizações,  
Parcial 2 Plotagem dos pontos de localizações em cima dos mapas  
Parcial 3 Animação (da sequencia de localizações em cima do mapa)  
Final1: Gif ou vídeo da movimentação do indivíduo 1  
Finaln: Gif ou vídeo da movimentação do indivíduo n
```

Como Produto Final teremos os Gifs (e Vídeos) gravados na pasta definida

Plano B: “DistLoc” Distância entre localizações reais e localizações estimadas

Contextualização

O uso de rádio-colares com localizações via GPS representam um grande avanço em termos de número de localizações obtidas, precisão e regularidade na coleta de dados. No entanto, diversos fatores podem ocasionar lacunas nas observações, isto é, algumas localizações podem faltar ou conter erros por não terem sido corretamente estimadas por problemas no sinal, seja porque a posição de satélites era ruim naquele momento, seja porque o animal estava num local que não captava o sinal (e.g. tocas, cavernas, etc). Por esse motivo foram criados alguns métodos capazes de estimar estes pontos faltantes. Alguns desses métodos são probabilísticos. Outros, podem simplesmente estimar uma linha reta para unir pontos. Outros ainda, podem permitir a utilização de modelos como GAM para estimar este ponto através de uma trajetória provável, estimada a partir de modelos. Em um dos capítulos do doutorado eu desenvolvi modelos de movimento controlando parâmetros dos mesmos e utilizei então os “biased random walks” gerados para estimar a precisão de modelos probabilísticos baseados em movimentos brownianos. No entanto, não cheguei a comparar métodos que não se baseiam na estimativa de probabilidades. A idéia para um projeto aqui, seria realizar um procedimento semelhante para um dos métodos não probabilísticos através de estimativas para localizações faltantes no package adehabitatLT ou utilizando um modelo GAM. Basicamente o procedimento consistiria em remover 1 a 1 pontos consecutivos de um “biased random walk” e estimar a distância de pontos estimados pelos adehabitatLT ou modelo GAM em relação a localização real que foi removida.

Objetivo: Criar uma função capaz de 1) remover localizações de uma random walk, 2) estimar os pontos fatantes através do adehabitatLT ou modelo GAM, 3) calcular as distâncias entre os pontos reais removidos e as estimativas do métodos citados, e 4) compilar os resultados das estimativas de diferentes trajetórias num arquivo txt que possa ser comparada com o resultado final de outro método num momento futuro.

Nome da função: DistLoc

Dados de entrada: a) dataframe com dados de movimento “biased random walks”

a) Dataframe

- id = individual.local.identifier, corresponde ao número de identificação única do indivíduo
- long = x, posição espacial no eixo x com base na longitude
- lat = y, posição espacial no eixo y com base na latitude

Pseudo-código:

1. Item de lista ordenada Cria objeto1 dando nome ao dataframe.
2. Monta ciclo para remoção sequencial com devolução das localizações removidas anteriormente
3. Estimativa do ponto removido através de comando do adehabitaLT ou de um modelo GAM
4. Cálculo da distância mínima entre pontos estimados e os reais removidos
5. Retorno dos resultados num dataframe e então em um arquivo txt salvo numa pasta determinada

Saída: Conjunto de arquivo txts com estimativas de distâncias entre pontos removidos e estimados. Tais arquivos podem ser comparados com estimativas de outros métodos para comparar a precisão de cada método ao estimar pontos faltantes de uma trajetória.

Referências

Grimm, Volker; Railsback, Steven F. (2005). Individual-based Modeling and Ecology. Princeton University Press. p. 485. ISBN 978-0-691-09666-7.

```
if(nchar(system.file(package="adehabitaLT")) > 0) citation("adehabitaLT")
```

```
if(nchar(system.file(package="move")) > 0) citation("move")
```

```
if(nchar(system.file(package="maptools")) > 0) citation("maptools")
```

```
if(nchar(system.file(package="ggplot2")) > 0) citation("ggplot2")
```

Yihui Xie (2013). animation: An R Package for Creating Animations and Demonstrating Statistical Methods. Journal of Statistical Software, 53(1), 1-27. URL <http://www.jstatsoft.org/v53/i01/>.

Calabrese JM, Fleming CH, Gurarie E. Ctm: an R Package for Analyzing Animal Relocation Data As a Continuous-Time Stochastic Process. Methods Ecol Evol. 2016.

Olá Alan,

Muito legal suas propostas, acho que você foi claro e ambas as propostas representam desafios compatíveis com o esperado nessa fase do curso. Mesmo que ambas as propostas estejam inseridas no contexto da área que você trabalha, você explica muito bem como elas abordam soluções não existentes para usuários dessa comunidade acadêmica e acho isso valioso. Pessoalmente gostei mais da proposta A, achei muito legal você sumarizar os resultados a um GIF. A parte visual no contexto da Ecologia do Movimento é muito importante porque senão se torna muito abstrata, como a proposta B por exemplo, onde sinto que uma saída gráfica faltou. Voltando à proposta A, não ficou claro para

mim que tipo de coordenadas você usaria nas colunas long e lat do df de entrada, se você permitiria vários formatos e depois transformaria dentro da função ou se exigiria um formato específico. Se o último for o caso, seria legal a função testasse isso e retornasse uma mensagem de erro caso os formatos das coordenadas não fosse o adequado. Outra mensagem de erro seria interessante se o mapa de entrada tem dimensões diferentes daquelas especificadas pelo df? Finalmente, não acho conveniente incluir a instalação dos pacotes necessários dentro da função porque isso significaria que a cada vez que a função é chamada ela baixará e instalará os pacotes de novo. Isso pode ser um problema para um usuário sem acesso a internet por exemplo. Uma solução seria que você indicasse no help da função os pacotes que devem ser instalados antes de rodar a função, a qual só os carregaria. Isso pode também poupar tempo de processamento. — [Gustavo Agudelo](#) 2018/05/10 10:53

— [Alexandre Adalardo de Oliveira](#) 2018/05/10 13:49

Oi Alan, Concordo com o Gustavo, fique com a proposta 1. Dê uma formatada melhor na documentação da sua função quando fizer, não sei se houve algum bug mas a formatação de alguns item ficam difíceis de entender parece que entrou um “item de lista não ordenada” em muitos pontos da sua redação... Gostei das sugestões do Gustavo, apenas discordo da instalação do pacote. Acho que deve incluir um teste lógico para verificar se os pacotes estão instalados, não estando, direcionar para ou: 1. uma mensagem de erro (o próprio require() faz isso) ou 2. dar a opção para o usuário instalar o pacote a partir de comando interativo (veja a função readline()) em que o usuário responde se deseja ou não instalar... Vá com a 2!

From:

<http://ecor.ib.usp.br/> - **ecoR**

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2018:alunos:trabalho_final:alanbiology:start



Last update: **2020/08/12 06:04**