

# Rodolfo Pereira Graciotti



Aluno de mestrado do departamento de Ecologia do IBUSP.

## Tabela de exercícios

[Exercícios - Rodolfo Pereira Graciotti](#)

## Laboratório

[Laboratório de Macroevolução e Macroecologia](#)

## Propostas de trabalho final

Olá Rodolfo,

Sou Gustavo A. Ballen, monitor encarregado de dar um retorno sobre suas propostas. Vou comentar em geral aqui sobre as duas propostas.

Acho a proposta A legal, só me preocupa um pouco usar os LADs e FADs tão estritamente mas tem vezes que não tem mais como incorporar informação temporal nessas análises. Se der, por quê não tentar modelar a incerteza sobre esses valores “observados” de tempo? Se bem não é essencial para a sua proposta, acho legal tentar pensar nesses detalhes, ainda mais porque é uma coisa que faz parte do seu plano de pesquisa na vida real. Eu

sugiro além disso tentar simular a incerteza sobre tal coexistência, já que todas as fontes de dados de entrada apresentam um tipo ou outro de incerteza. Dado que você é um usuário experiente do R e domina técnicas de simulação e programação, acredito que do jeito que esta a função não é muito desafiante para você, porém incorporar alguma maneira de simular incerteza irá acrescentar bastante a proposta.

A proposta B é simplesmente uma calculadora encapsulada numa função, nem vale a pena considerá-la uma alternativa.

Por gentileza rever as sugestões antes de aceitarmos a proposta para o trabalho final.

[Gustavo A. Ballen](#)

## Proposta A - Co-ocorrência de organismos fósseis

### Contextualização

A paleontologia é um ramo da ciência que lida com grande quantidade de incertezas. A natureza do registro fóssil é incompleta, portanto algumas suposições e simplificações às vezes se tornam necessárias. Estudos de distribuição de espécies extintas tornam-se limitados à qualidade dos dados, mas são passos importantes para compreender processos ecológicos em tempo profundo. Determinar a coexistência de espécies no tempo e no espaço não é uma tarefa simples, mas alguns mecanismos já foram propostos para determinar graus de simpatria entre espécies fósseis <sup>1)</sup>. Um ponto de partida é trabalhar com ocorrências fósseis geo-referenciadas das espécies que se busca estudar. Uma ocorrência fóssil é o registro espaço-temporal de um indivíduo de uma espécie (geralmente extinta, mas podemos ter fósseis de organismos que ainda estão vivos).

Há um esforço coletivo da comunidade paleontológica de compilar as ocorrências fósseis provenientes de museus em grandes bancos de dados, a exemplo da [Paleobiology Database](#) e da [NOW](#). Essas bases de dados são amplamente utilizadas em trabalhos que aliam ferramentas estatísticas aos dados paleontológicos, a fim de se responder questões ecológicas em tempo profundo <sup>2)</sup>.

Partindo das ocorrências fósseis compiladas nas coleções museológicas, minha proposta de função visa identificar pares de espécies que potencialmente coexistiram no **tempo** e no **espaço**, gerando uma matriz de coexistência entre as espécies de um determinado grupo de interesse. Dessa forma, será possível investigar potenciais efeitos de interações ecológicas.

### Planejamento da função

#### Entrada

O dado de entrada deverá ser um objeto de data . frame, com as ocorrências fósseis como linhas. Dada a natureza do registro fóssil, cada ocorrência fóssil não pode ser datada diretamente, mas sim

alocada a um ou mais possíveis intervalos estratigráficos. Essa incerteza deve estar incorporada no `data.frame` e poderá ser trabalhada de diferentes formas. Essa incerteza é representada com duas colunas, uma com a data de início do intervalo estratigráfico, e uma com o final do intervalo estratigráfico, ambas na escala de milhões de anos. Cada ocorrência deverá conter também o nome da espécie, as coordenadas geográficas e um identificador de sítio de coleta. O sítio de coleta deverá ser algo que identifica a assembléia fóssil a qual esse organismo foi escavado, e é único para cada assembléia, pois o mesmo ponto geográfico de amostragem no espaço pode não estar associado com a mesma camada estratigráfica, ou seja, não está associado ao mesmo intervalo de tempo dada a natureza vertical das camadas bioestratigráficas. Essa informação geralmente está associada à coleção em que aquela ocorrência fóssil está catalogada no museu e é facilmente recuperável nos bancos de dados como o [Paleobiology Database](#) e o [NOW](#). Dessa forma, a tabela de entrada conterá **ocorrências** fósseis de todas as espécies do grupo de interesse, sendo que uma espécie poderá ter diversas ocorrências, ou apenas uma, caso seja conhecido um único fóssil daquela espécie.

Exemplo:

sp	sítio	longitude	latitude	base do intervalo	topo do intervalo
Dinossaurus terrificus	1	-104.2	40.8	175.4	168.2
Dinossaurus horrendus	1	-104.2	40.8	175.4	168.2
Dinossaurus terrificus	2	-85.6	37	175.4	168.2
Dinossaurus magnificus	3	-104.2	40.8	155.4	147.3
Dinossaurus magnificus	4	-107.8	50.5	155.4	147.3

## Argumentos

Além do dado de entrada, planejo incluir argumentos que definem as formas como o usuário pode lidar com incertezas e qual o método será utilizado para calcular a sobreposição das espécies.

**Incerteza temporal:** O usuário deve alocar cada ocorrência a um único intervalo, utilizando o ponto médio da estimativa de idade daquela ocorrência, ou a base do intervalo. Para isso, o usuário pode entrar com um vetor de intervalos estratigráficos conhecidos que se busca alocar as ocorrências, ou criar intervalos próprios, por exemplo a cada 10 milhões de anos (pode ser o default). Esse primeiro passo é essencial para a determinação da sobreposição no **tempo**.

- Exemplo de vetor de tempo: 37.2 33.9 30.3 20.44 ...

**Métodos espaciais:** Proponho 3 métodos para identificar a sobreposição no **espaço**, que o usuário poderá escolher e serão elaborados na descrição da função: `sítio`, `poligono` e `distância`.

Exemplo:

```
time_space(df, alocacao, intervalos, metodo espacial)
```

Em que `df` é o `data.frame` de entrada, `alocacao` determina se as ocorrências serão alocadas à base ou ao ponto médio do intervalo, `intervalos` é o vetor de intervalos estratigráficos ou cortes de tempo, `metodo espacial` é a determinação do método que será usado para a sobreposição espacial.

## Pseudo-código

### Pacotes

- Serão necessários alguns pacotes externos para o cálculo dos métodos espaciais (como `geosphere`, `fossil`, `sp`), essa etapa irá checar se estão instalados e, se necessário, instalá-los e posteriormente carregá-los.

### Checagem

1. Essa seção identificará se o conjunto de dados foi entrado corretamente, isto é, se contém as colunas corretas, com os nomes corretos e dimensões corretas; utilizando-se de testes lógicos.
2. Determinação dos métodos de alocação e espacial com testes lógicos.
3. Exemplo: `if(alocacao == "midpoint" & intervalos == "vetor"){loop que checa a qual intervalo pertence aquela ocorrência e identifica uma a uma com uma nova coluna}`

### Tempo

1. Cria um loop para determinar, a cada intervalo de tempo, quais espécies estão presentes naquele intervalo
2. Recolhe do loop uma lista para cada intervalo de tempo com os pares de espécies que coexistem
3. Cria uma matriz de "distância" entre todas as espécies, com `TRUES` se há coexistência em pelo menos um momento no tempo

### Espaço

Dada a seleção do usuário, partirá para um dos 3 caminhos possíveis, cada caminho dentro de um `if`

### Sítio

A sobreposição por sítios é a evidência mais direta de sobreposição, pois aqui podemos identificar casos em que fósseis de espécies diferentes foram encontrados juntos, no mesmo local de coleta, que reflete o mesmo momento no tempo e no espaço.

1. Determina quais sítios possuem mais que uma espécie
2. Cria uma lista com todos os sítios e identifica para aqueles que possuem mais que uma espécie, quais as espécies presentes
3. A partir da lista, cria uma matriz de "distância" entre todas as espécies, com `TRUES` se há coexistência em pelo menos um sítio

### Polígono

A construção de mínimos polígonos convexos para espécies fósseis é amplamente utilizada na inferência da área geográfica dessas espécies<sup>3)</sup>, portanto, uma forma de medir coexistência é identificar a sobreposição dos polígonos das espécies em determinados momentos de tempo.

1. Separa as ocorrências de cada espécie a cada intervalo de tempo em uma lista
2. Checa se cada espécie em cada fatia de tempo possui pelo menos 3 pontos para que seja possível o cálculo do polígono. Espécies que não atendem esse critério serão descartadas
3. Aplica funções que calculam o mínimo polígono convexo para cada seção da lista, com funções oriundas dos pacotes `fossil` e/ou `sp`
4. Calcula se há sobreposição de polígonos de espécies a cada momento do tempo, e identifica esses casos, novamente com funções dos pacotes acima
5. Retorna uma matriz final de “distância” entre todas as espécies, com `TRUES` se há sobreposição dos polígonos das espécies em pelo menos um momento de tempo

### Distâncias

Uma alternativa aos métodos dos polígonos, que requer uma quantidade maior de ocorrências por espécie, é calcular as distâncias médias entre os pontos das espécies e determinar um limiar de coexistência, isto é, se os pontos mais próximos de duas espécies diferentes encontram-se a uma distância menor que o limiar, será determinado que essas espécies coexistiram. Para isso, é necessário determinar esse limiar a partir da distância média entre os pontos de cada espécie, e determinar uma média geral dessas distâncias das espécies. A ideia é que essa distância média total reflete uma capacidade de dispersão média das espécies do grupo de interesse, e que espécies que possuem menor capacidade de dispersão possuirão menores distâncias médias entre todos os pontos.

1. Separa as ocorrências de cada espécie a cada intervalo de tempo em uma lista
2. Checa se cada espécie em cada fatia de tempo possui pelo menos 2 pontos para que seja possível o cálculo das distâncias. Espécies que não atendem esse critério serão descartadas
3. Aplica a função `distm` do pacote `geosphere`, que calcula matrizes de distância para cada elemento da lista
4. Retorna o valor da média para cada elemento da lista e simplifica em um vetor (`sapply`)
5. Sobrescreve valores de distância 0 para `NA` (isso ocorre quando espécies possuem todos os pontos amostrado na mesma localidade)
6. Calcula a média geral
7. Calcula se a menor distância entre pares de pontos de espécies diferentes é menor que a média geral
8. Retorna uma matriz final de “distância” entre todas as espécies, com `TRUES` se a menor distância entre duas espécies é menor que a distância média em pelo menos algum momento de tempo

### Multiplicação das matrizes

- Multiplica a matriz de coexistência temporal com a matriz de coexistência espacial escolhida e a guarda em um novo objeto. Assim como as matrizes anteriores, conterá todas as espécies do grupo de interesse como linhas e colunas, e será preenchida com `TRUE` onde foi determinada a coexistência espaço-temporal, e `FALSE` para pares de espécies sem nenhuma evidência de coexistência espaço-temporal.

## Saída

- Retorna a matriz final de coexistência espaço-temporal.
- 

# Proposta B - Calculadora de chance de captura de pokémon

## Contextualização

A série de jogos Pokémon é extremamente popular, desde sua criação nos anos 90 até hoje. Apesar de superficialmente simples e apelativos para um público infantil, os jogos da série possuem também um apelo para jogadores “hardcore”, que buscam uma experiência de jogo mais imersiva. Diversas mecânicas dos jogos já foram exploradas a fundo pela comunidade de jogadores. Os pokémons são o principal “coleccionável” dos jogos, em que o jogador pode “capturar” diversos pokémons, de “espécies” diferentes, montar o seu time de pokémons e colocá-los para se enfrentar em batalhas por turnos. Dentre uma das principais mecânicas dos jogos, destaca-se a captura de pokémons. Pokémons possuem probabilidades diferentes de captura, refletindo de certa forma a sua natureza (raro, comum, único, etc.) e um mecanismo bastante complexo determina a probabilidade do jogador obter um pokémon a cada tentativa de captura. O cálculo dessa probabilidade depende de uma série de variáveis, e variou ao longo da série de jogos, sendo mais simples em seu início e avançando em complexidade até os jogos mais recentes.

O cálculo da mecânica de captura não é acessível ao jogador diretamente, e há um grande esforço da comunidade de jogadores em documentar como isso é feito. O fórum online [Bulbapedia](#) é mantido por fãs e contém uma infinidade de documentações sobre como o jogo funciona, sendo uma valiosa fonte de informação para os jogadores mais ávidos. Algumas ferramentas para o cálculo da taxa de captura já foram propostas em sites e fóruns da internet, baseando-se na documentação já disponível na [Bulbapedia](#).

Minha proposta é criar uma ferramenta em R que permite o usuário calcular a chance de captura de um pokémon em um evento de batalha, baseando-se nas equações descritas na [Bulbapedia](#). Dessa forma, o jogador poderá saber qual a chance que terá de capturar um pokémon a cada evento de captura, dando as variáveis corretas à função. Cabe ressaltar que proponho algumas simplificações para a proposta, como programar a função para alguns jogos apenas, dada a grande possibilidade de variações no cálculo conforme o jogo selecionado; programar apenas algumas classes de pokébolos (que são modificadores da chance de captura); desconsiderar modificadores de jogos muito recentes que ainda não são muito bem descritos pela comunidade, etc.

## Planejamento da função

### Entrada

O objeto de entrada corresponderá a um conjunto de argumentos que descreverão o jogo selecionado, o pokémon selecionado, e uma série de variáveis que serão universais para todos os

jogos:

1. **jogo**: O usuário poderá escolher dentre as opções qual o jogo utilizado, para que então a função possa calcular a probabilidade conforme a equação correta, descrita para cada jogo. Esse argumento é da classe `character`.
2. **pokemon**: O usuário deverá identificar o pokémon que se deseja capturar. Cada pokémon tem um valor constante para todos os jogos que será utilizado nas equações. Como cada pokémon é identificado com um número de 001 a 809, essa entrada será da classe `numeric`.
3. **pokebola**: Pokébolas são modificadores da chance de captura, para fins de simplificação, o usuário poderá escolher dentre apenas 3 possibilidades. Essa entrada é da classe `character`.
4. **HP max**: Cada pokémon encontrando no jogo terá um valor de `HP max`, que seria o nível de dano que ele pode receber durante a batalha. Se chegar a 0, a batalha é terminada sem que o jogador tenha a chance de capturá-lo. Esse valor é indicado no começo da batalha e deve ser da classe `numeric`.
5. **HP atual**: Ao longo da batalha, o objetivo é reduzir o HP para o menor valor possível (antes de chegar a 0), pois esse valor é um multiplicador do cálculo, e deverá ser incluído na função no momento em que se deseja iniciar as tentativas de captura. Deverá se também da classe `numeric`.
6. **status**: Ao longo da batalha, também é possível infligir um "status" no pokémon, que reduz a sua capacidade de combate e também serve como um modificador do cálculo de captura. Deverá ser da classe `character`.

## Pseudo-código

### Checagem e criação de vetores

1. O primeiro passo é criar os vetores de valores de base para cada pokémon identificado com o seu número, de acordo com a lista compilada [aqui](#) <sup>4)</sup>.
2. Cria-se um vetor com todos os valores associados aos modificadores de pokébolas: `if(pokebola == "pokeball"){bonusball ← 1} (default); if(pokebola == greatball){bonusball ← 1.5}`, etc.
3. Cria-se um vetor com todos os valores associados aos modificadores de status: `if(status == "sleep"){bonusstatus ← 2}`, `if(status == "none"){bonusstatus ← 1} (default)` etc.

### Determinação do jogo

- Outro passo crucial é determinar qual a equação que será utilizada, dada as possibilidades de jogo: `gen_I`, `gen_II`, ou `G0`, que correspondem à primeira, segunda gerações de jogos, e o jogo 'Pokémon GO', respectivamente, etc.

### Aplicando equação de jogo

- Para cada jogo selecionado, haverá uma fórmula diferente e potenciais complicações, mas a estrutura básica é similar ao exemplo:

$$a = \frac{(3 \times \text{HP}_{\max} - 2 \times \text{HP}_{\text{current}}) \times \text{rate} \times \text{bonus}_{\text{ball}}}{3 \times \text{HP}_{\max}} \times \text{bonus}_{\text{status}}$$

- $HP_{max}$  é o HP do pokémon no início da batalha,
- $HP_{current}$  é o HP do pokémon no momento da tentativa de captura,
- $rate$  é a taxa base de captura de cada pokémon,
- $bonus_{ball}$  é o modificador de pokébola,
- $bonus_{status}$  é o modificador de status.
- O objeto de saída da equação será o valor da taxa de captura  $p$ .

### Saída

- O objeto final da equação será o valor de  $p$ , a taxa de captura naquele momento para o pokémon, jogo, e condições selecionadas.
- 

## Trabalho final

Para o trabalho final, decidi desenvolver a proposta A, a função de coexistência espaço-temporal de linhagens fósseis: `coex_st`. De acordo com as sugestões, implementei um método que lida com a incerteza da estimativa de surgimento e extinção das linhagens no registro fóssil. A função final difere em algumas metodologias da proposta original, tornando-a mais coesa. O objeto final da função é uma série temporal de coexistência, definida por uma lista de matrizes binárias 0/1 para determinar a não-coexistência/coexistência. Os métodos de coexistência espacial diferem um pouco da proposta, sendo possível 3 opções, com diferentes graus de permissividade: sítios de coleta, coordenadas geográficas absolutas e polígonos. Incluí também um parâmetro que determina um output gráfico, que explora propriedades da função. Detalhes no help da função. O exemplo da função necessita do `data.frame` disponibilizado abaixo.

- Link para o código da função: [Função coex\\_st](#)
- Link para download da função em `.r`: [coex\\_st.r](#)
- Link para a página de ajuda: [Help](#)
- Link para download da página de ajuda como `.txt`: [Help\\_coex\\_st.txt](#)
- Link para download do `data.frame` necessário para rodar o exemplo da função: [canidae\\_data\\_frame.csv](#)

1)

[Darroch, S., & Saupe, E. \(2018\)](#) , [Knapp et al. \(2018\)](#)

2)

[Silvestro et al. \(2015\)](#) , [Pires et al. \(2017\)](#)

3)

[Darroch, S., & Saupe, E. \(2018\)](#)

4)

[https://bulbapedia.bulbagarden.net/wiki/List\\_of\\_Pok%C3%A9mon\\_by\\_catch\\_rate](https://bulbapedia.bulbagarden.net/wiki/List_of_Pok%C3%A9mon_by_catch_rate)

From:

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

Permanent link:

[http://ecor.ib.usp.br/doku.php?id=05\\_curso\\_antigo:r2019:alunos:trabalho\\_final:rodolfo.graciotti:start](http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2019:alunos:trabalho_final:rodolfo.graciotti:start) 

Last update: **2020/09/23 17:15**