

Beatriz Foganholi Fernandes

Também conhecida como Brabuleta (ou Brabs)!



1. Quem sou? O que faço?

Fiz Bacharelado e Licenciatura em Ciências Biológicas no Instituto de Biociências da USP (IB-USP). Sou mestranda no Departamento de Fisiologia Geral neste mesmo Instituto, trabalhando no Laboratório de Metabolismo Energético e Sazonalidade, sob orientação das Profas. Dras. Silvia Cristina Ribeiro de Souza e Gisele Akemi Oda.

Gosto muito de Cronobiologia e o interesse do meu mestrado é investigar os efeitos do ciclo ambiental de temperatura em aspectos fisiológicos de um animal ectotérmico, a rã touro americana. O título do meu projeto, mais especificamente, é: “Efeitos de altas temperaturas associadas a termociclo diário sobre a atividade e a imunocompetência em *Lithobates catesbeianus* (Shaw, 1802) em verão e em inverno.”.

2. Exercícios resolvidos durante o curso

Venha conferir meus exercícios resolvidos aqui ⇒ [Exercícios da Brabuleta](#).

3. Trabalho final: Criando uma função

3.1. Proposta A: Criando um actograma!

Contextualização

Você já ouviu falar nos ritmos biológicos?

Os ritmos biológicos estão amplamente distribuídos entre os seres vivos conhecidos, sendo observados tanto em seres unicelulares quanto em multicelulares. Consistem em flutuações cíclicas de variáveis biológicas nos mais diversos níveis de organização, desde a expressão de um gene, a secreção rítmica de um hormônio, o comportamento (como o padrão de alternância entre sono e vigília em animais) ao nível populacional (como os padrões de migrações sazonais). Como resultado, tem-se a distribuição temporal das diversas funções biológicas ao longo do dia e da noite, do mês ou do ano. A figura abaixo traz um exemplo de um ritmo de movimentação foliar observado em *Phaseolus multiflorus*. [Mais informações [aqui!](#)]:



Essa distribuição temporal das funções muitas vezes acontece de forma associada às variações ambientais, como de dia e noite, quente e frio. No entanto, essa relação nem sempre é obrigatória! É possível que um ritmo biológico se mantenha mesmo que todas as variações do ambiente sumam. Por esse motivo, os estudos dos ritmos biológicos, feitos pela chamada Cronobiologia, acontecem associados a diferentes condições ambientais.

Uma ferramenta muito utilizada para analisar padrões de ritmos na Cronobiologia é o actograma, que consiste num empilhamento das atividades registradas dia a dia, com o objetivo de permitir a visualização de padrões na escala diária. Para facilitar ainda mais essa visualização, geralmente são feitos na forma de dupla. Imagine que você tem registros da movimentação de um animal durante 120 horas corridas, e que, se você desenhasse essa movimentação numa fita, ela estaria marcado em rosa como na figura abaixo. Para construir um actograma, basta “cortar” sua fita em pedaços de 24 horas e colocá-las empilhadas em sequência, indo do dia A ao dia E. Por fim, basta duplicar essas fitas e alinhá-las com sua vizinha seguinte para formar um actograma duplo, criando pares AB, BC, CD e DE. como consequência dessa cópia deslocada, a primeira e a última linha do actograma ficam sem par.



Um exemplo mais real de um actograma, para diferentes condições de claro-escuro ambiental, pode ser visto [Neste trabalho com camundongos](#) em que áreas cinzas do gráfico são utilizadas para sinalizar os momentos de escuro, enquanto as áreas brancas sinalizam os momentos de claro. As demarcações em preto sinalizam os momentos de atividade do camundongo, andando na roda:



Como esta é uma ferramenta visual muito utilizada na análise dos ritmos biológicos, ao mesmo tempo em que não é disponibilizada por nenhum programa gratuito, minha proposta é desenvolver uma função para plotagem de actogramas a partir de dados de ritmos biológicos.

Planejamento da Função

Entrada

actograma(hora, ritmo, n.dias, Ei, Ef, Fi, Ff)

- hora: Conjunto de dados referente aos horários dos dias em que o ritmo foi coletado.
- ritmo: Conjunto de dados referente a amplitude do ritmo que foi coletado.
- n.dias: Total de dias desejados no actograma.
- Ei: vetor contendo a(s) hora(s) de início do escuro, repetida pelo número de dias da(s) respectiva(s) condição(ões). Inclui a possibilidade constante.
- Ef: vetor contendo a(s) hora(s) de fim do escuro, repetida pelo número de dias da(s) respectiva(s) condição(ões). Inclui a possibilidade constante.
- Fi: vetor contendo a(s) hora(s) de início do frio, repetida pelo número de dias da(s) respectiva(s) condição(ões). Inclui a possibilidade constante.
- Ff: vetor contendo a(s) hora(s) de fim do frio, repetida pelo número de dias da(s) respectiva(s) condição(ões). Inclui a possibilidade constante.

Verificação de parâmetros

- hora e ritmo possuem o mesmo comprimento? Se não, retorna "'hora' e 'ritmo' possuem comprimentos distintos."
- hora deve conter valores no formato de hora. Se não, retorna "Dados contém 'hora' fora de formato".
- n.dias é número inteiro positivo? Se não, retorna "'n.dias' deve ser número inteiro e positivo."
- Ei e Ef devem estar no formato de hora. Se não, retorna "Regime(s) de escuro fora de formato".
- Fi e Ff devem estar no formato de hora. Se não, retorna "Regime(s) de frio fora de formato".
- Ei e Ef devem ter o mesmo comprimento. Se não, retorna "'Ei' e 'Ef' possuem comprimentos distintos".
- Fi e Ff devem ter o mesmo comprimento. Se não, retorna "'Fi' e 'Ff' possuem comprimentos distintos".

Pseudocódigo

1. Subdividir o conjunto de dados ritmo a cada dia, usando meia-noite como ponto de quebra em hora;
2. Criar uma lista amplitudes para as subdivisões diárias de ritmo a partir dessa quebra;
3. Criar as listas E.inicio, E.fim, F.inicio, F.fim para armazenar as repetições inseridas nos argumentos Ei, Ef, Fi e Ff respectivamente;
4. Definir um objeto ylim com os valores mínimo e máximo contidos no conjunto ritmo;
5. Criar uma matriz layout para a plotagem do actograma, com n.dias+1 linhas e 2 colunas;
6. Criar um ciclo for com contador [i] de 2: (n.dias+1) para plotagem de histogramas na primeira coluna do layout, pulando o primeiro espaço da coluna:
 1. Plota o histograma dos dados contidos em amplitudes[i];
 2. Plotagem do ciclo claro-escuro com uso de cláusulas se:
 1. Para claro constante, Ei = Ef = 0, não cria nenhum polígono;
 2. Para escuro constante, cria retângulo cinza com y0 = min(ylim), y1 = max(ylim), x0 = Ei = 0, x1 = Ef = 23:59;
 3. Para presença de claro e escuro:
 1. Se a fase de escuro inclui a meia-noite:
 1. Cria dois retângulos cinzas, com y0 = min(ylim), y1 = max(ylim):
 1. x0 = meia-noite, x1 = E.inicio[i] e
 2. x0 = E.fim[i], x1 = meia-noite.
 2. Se a fase de escuro não inclui a meia-noite:
 1. Cria retângulo cinza com y0 = min(ylim), y1 = max(ylim), x0 = E.inicio[i] e x1 = E.fim[i].
 3. Plotagem do ciclo quente-frio com uso de cláusulas se:
 1. Para temperatura constante, Fi = Ff = 0, mantém o histograma de amplitudes em preto.
 2. Para ciclo de temperatura presente, cria cortes no histograma utilizando os pontos Fi e Ff, que delimitam coloração do histograma preenchida em azul e mantém o restante com coloração preenchida em vermelho.
7. Criar um ciclo for com contador [i] de 1:n.dias para plotagem de histogramas na segunda coluna do layout, mantendo em branco o último espaço da coluna, repetindo todas as etapas do item 6.
8. Criar o objeto num.dias com a mensagem "Actograma duplo criado com dados de n.dias

dias”;

9. Criar o objeto CE em forma de tabela com as colunas Duração do claro, Duração do escuro e Número de dias, e as linhas com os regimes utilizados;
10. Criar o objeto QF com a mensagem “Regime de temperatura constante” ou em forma de tabela com as colunas Duração do quente, Duração do frio e Número de dias, e as linhas com os regimes utilizados;
11. Retorna lista com os objetos num.dias, CE e QF.

Saída

- Gráfico do actograma em duplo-plot;
- Lista no terminal contendo:
 - Número de dias contidos no gráfico;
 - regime(s) ambiental(is) de claro-escuro (CE) e
 - regime(s) ambiental(is) de quente-frio (QF).

3.2. Proposta B: Utilizando uma tabela de vida!

Contextualização

Quando se trata de qualquer estudo em Biologia, conhecer o ciclo de vida de uma espécie de interesse parece ser um dos primeiros passos antes de aprofundar qualquer investigação. Estudos demográficos, que monitoram, por exemplo, sobrevivência, mortalidade e fecundidade de populações nos permitem fazer previsões e inferências temporais. Além disso, se tornam uma ferramenta para entender efeitos externos, como de doenças ou de alterações ambientais. Esses monitoramentos demográficos podem ser feitos por meio das chamadas tabelas de vida. Um exemplo simplificado pode ser visto na [tabela](#) a seguir, que traz a contabilização de sobrevivência e fecundidade de pássaros numa ilha, tanto em anos de chuva como de seca:



As tabelas de vida podem ser feitas de duas formas:

- Quando somos capazes de acompanhar o envelhecimento de uma coorte, que diz respeito ao conjunto de indivíduos que nasceu numa mesma faixa de tempo. Essa é a chamada Tabela de vida dinâmica (ou de coorte).
- Quando observamos indivíduos de coortes distintas e que coexistem num mesmo período, sendo capazes de diferenciar suas idades. Essa é a chamada Tabela de vida estática (ou de tempo específico).



Como esta é uma ferramenta que pode ter uso bastante amplo nas diferentes áreas da biologia - como em conhecer sua espécie de estudo, investigar efeitos ambientais, acompanhar efeitos de medicamentos ou puramente auxiliar em planejamentos metodológicos -, minha proposta é desenvolver uma função que retorne uma tabela de vida completa a partir da escolha do usuário, além de criar gráficos para visualização dos dados de interesse e de curvas demográficas deles

derivadas.

Planejamento da Função

Entrada

```
tab.vida(classes, vivos, fecund = FALSE, tipo)
```

- `classes`: Conjunto de dados referente às classes de idade da espécie de interesse, podendo ser um vetor numérico ou de fatores.
- `vivos`: Conjunto de dados referente ao número de indivíduos vivos observados em cada classe.
- `fecund`: Conjunto de dados contendo os valores de fecundidade observados por classe, presente apenas no `tipo = din`.
- `tipo`: Argumento que define que tipo de tabela de vida o usuário deseja usar, podendo ser dinâmica (`din`) ou estática (`est`).

Verificação de parâmetros

- Os vetores `classes` e `vivos` possuem o mesmo comprimento? Se não, retorna "'classes' e 'vivos' apresentam comprimentos distintos".
- O vetor `vivos` contém valores inteiros positivos? Se não, retorna "'vivos' deve ser de números inteiros e positivos".
- O vetor `fecund`, quando presente, contém valores inteiros positivos? Se não, retorna "'fecund' deve ser de números inteiros e positivos".
- * O vetor `fecund`, quando presente, é de mesmo comprimento que `classes` e `vivos`? Se não, retorna " Verifique o comprimento de 'fecund'".
- `tipo` está recebendo `din` ou `est`? Se não, retorna "Escolha entre tabela dinâmica (`din`) ou estável (`est`)".

Pseudocódigo

1. Se `tipo == din`:
 1. Transformar o conjunto de dados em um *dataframe* `vida`, contendo as colunas `classes`, `vivos` e `fecundidade`;
 2. Adicionar uma coluna `vivos.p` com os valores padronizados de sobreviventes considerando-se a primeira classe etária como 1000;
 3. Adicionar uma coluna `mortos.p` a partir da diferença no número de sobreviventes entre uma classe etária e a seguinte;
 4. Adicionar uma coluna `taxa.mort` a partir da divisão `mortos.p/vivos.p`;
 5. Adicionar uma coluna `log.ax` com o log, na base de 10, dos valores de `vivos`;
 6. Adicionar uma coluna `poder.let` a partir do cálculo $(\log.ax[x]) - (\log.ax[x+1])$;
 7. Criar dispositivo para 4 gráficos conjuntos
 1. Plotar histograma do valor absoluto de `vivos` em função de `classes`;
 2. Criar um gráfico de linha de `fecund` em função de `classes` para ilustrar fecundidade;
 3. Criar um gráfico de linha de `vivos.p` em função de `classes` para ilustrar sobrevivência;

4. Criar um gráfico de linha de taxa.mort em função de classes para ilustrar taxa de mortalidade;
 8. Retornar o *dataframe* vida.
2. Se `tipo == est`:
1. Transformar o conjunto de dados em um *dataframe* vida, contento as colunas classes e vivos;
 2. Adicionar uma coluna vivos.p com os valores padronizados de sobreviventes considerando-se a primeira classe etária como 1000;
3. Adicionar uma coluna mortos.p a partir da diferença no número de sobreviventes entre uma classe etária e a seguinte;
1. Adicionar uma coluna taxa.mort a partir da divisão mortos.p/vivos.p;
 2. Adicionar uma coluna taxa.vivo a partir da divisão vivos.p/1000;
 3. Criar dispositivo para 4 gráficos conjuntos
 1. Plotar histograma do valor absoluto de vivos em função de classes;
 2. Criar um gráfico de linha de taxa.vivo em função de classes para ilustrar taxa de sobrevivência;
 3. Criar um gráfico de linha de taxa.mort em função de classes para ilustrar taxa de mortalidade;
 4. Retornar o *dataframe* vida.

Saída

- Na escolha de tabela de vida dinâmica ou de coorte:
 - Tabela de vida a partir dos dados apresentados, contendo os números padronizados de vivos.p, mortos.p, taxa.mort e poder.let além dos dados informados inicialmente de vivos e fecund;
 - Dispositivo gráfico contendo:
 - Histograma do número de vivos observados por classe de idade;
 - Curva de fecundidade;
 - Curva de sobrevivência;
 - Curva de mortalidade.
- Na escolha de tabela de vida estática ou de tempo específico:
 - Tabela de vida a partir dos dados apresentados, contendo os números padronizados de sobreviventes, mortos, proporção de sobrevivência e taxa de mortalidade a partir dos dados informados inicialmente de classe de idade e número de vivos por início de classe de idade;
 - Dispositivo gráfico contendo:
 - Histograma do número de vivos observados por classe de idade;
 - Curva de sobrevivência;
 - Curva de mortalidade.

Resultado do Trabalho Final: Proposta A

Não deixe de dar uma olhadinha na função que criei!:

- Confira o código da função `actograma()` aqui ⇒ [Função: actograma\(\)](#).
- Confira a página de ajuda da função `actograma()` aqui ⇒ [Ajuda: actograma\(\)](#).

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

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2019:alunos:trabalho_final:beatriz.foganholi.fernandes:start&rev=1597223093 

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