Jonathan W. Lawley

Biólogo, Mestrando em Zoologia pelo Instituto de Biociências da Universidade de São Paulo

Laboratório de Cultivo e Estudos de Cnidaria

Currículo Lattes: http://lattes.cnpq.br/3208016919493506
Contato: jonathan.lawley@yahoo.com.br | jwlawley@ib.usp.br

Trabalho com sistemática e biogeografia de *Aurelia* (Cnidaria, Scyphozoa), comumente cultivadas em aquários ao redor do mundo. Mais especificamente, estudo a delimitação e distribuição das espécies deste gênero.

O título do meu projeto é: “A identidade da água-viva *Aurelia* (Cnidaria, Scyphozoa) do litoral brasileiro e discussões sobre a sistemática do gênero”, orientado pelo Prof. Dr. André C. Morandini.

Meus Exercícios

Clique para a acessar meus exercícios resolvidos

Trabalho Final

Plano A: Teste de isometria e correção de tamanho para dados morfométricos

Contextualização

Para levar em consideração diferenças na forma, a morfologia deve ser caracterizada independente do tamanho. Sendo assim, ao comparar dados morfométricos pertencentes a indivíduos de diferentes tamanhos, deve haver uma correção para que todos estejam na mesma escala. Para tanto, há uma variável operacional de tamanho que serve de referência para a correção dos outros dados morfométricos medidos. Porém, a relação entre esta variável de referência e a(s) variável(is) resposta pode ser isométrica ou alométrica. Se a variação entre as duas for proporcional, seguindo a ordem de potência das variáveis medidas, a relação será isométrica, e se não, alométrica. Por exemplo, se medirmos dados de envergadura e altura de indivíduos, em centímetros, a relação entre eles será isométrica se o logaritmo desses dados plotados se adequarem a uma reta de inclinação 1/1, que são os expoentes das potências de centímetros. Em outro exemplo, se a relação fosse entre massa e altura, a inclinação da reta para isometria deveria ser 3/1, já que o expoente da potência do volume (massa) é 3. A correção dos dados se dá então pela fórmula demonstrada em Lleonart et al. (2000):

Nesta equação, *Yi** é a variável resposta corrigida do espécime *i*, enquanto que *Yi* é a variável resposta medida deste mesmo espécime. *X0* representa a média da variável referência no conjunto de dados, e *Xi*, o valor da variável referência para o espécime em questão. A inclinação da reta entre a variável resposta e referência medidas é representada por *b*. Se a relação entre as variáveis for isométrica, *b* será a inclinação da reta que segue a razão entre os expoentes das variáveis resposta e referência. Se a relação for alométrica, *b* será a inclinação da reta obtida pelo modelo linear entre os dados. Sendo a relação alométrica ou isométrica, a variável referência transformada será a média da variável referência observada, pois removemos a variação de tamanho. Para mais detalhes e exemplos veja Colton (1999).

Planejamento da função

Entrada: isometry.test (`response`, `reference`, `specimen.id`, `expected.slope`, `alpha`)

- `response` = variável(is) resposta medida(s) (classe: numeric).
- `reference` = variável de referência para correção, geralmente a variável operacional de tamanho (classe: numeric).
- `specimen.id` = nome dos espécimes medidos (classe: character).
- `expected.slope` = inclinação da reta esperada para isometria (classe: numeric).
- `alpha` = nível de significância a ser utilizado, expresso como probabilidade (classe: numeric, 0 < alpha < 1).

Verificando os parâmetros:

- `response` está presente, não é um fator e é coercível para numérico? Se não, para e escreve: “response should be specified as a numeric vector containing the response measurement(s) for isometry test”.
- `reference` está presente, não é um fator, é coercível para numérico e tem o mesmo número de espécimes que response? Se não, para e escreve: “reference should be specified as a numeric vector containing the reference measurements for isometry test”.
- `specimen.id` está presente, é um caráter e tem o mesmo número de espécimes que `reference`? Se não, para e escreve “specimen.id should be specified as a character vector containing the names of specimens in data table”.
- `expected.slope` está presente, é numérico e corresponde ao número de variáveis resposta inseridas? Se não, 1 é atribuído a todas as variáveis resposta e escreve “expected.slope should be a concatenation of the ratios: power of response variable/power of the reference variable; herein default set to 1 for all”.
- `alpha` está presente, é numérico e > 0 e < 1? Se não, 0.05 é atribuído a `alpha` e escreve “alpha should be numeric and >0 or <1; herein default set to 0.05”.

Pseudo-código:

1. Pede o nome da variável de referência e salva no objeto `ref.name` (para nomear os dados correspondentes na tabela final).
2. Cria o objeto `result` com um *data frame* de NAs de 1 linha e `response` colunas.
3. Cria o objeto `result2` com uma matriz de NAs de `response` linhas e `response` + `reference` colunas.

4. Atribui `reference` para a primeira coluna de `result2`.
5. Se só há uma variável resposta, a função pede o nome dessa variável e salva no objeto `res.name` (para nomear os dados correspondentes na tabela final).
6. Cria o objeto `response.all` com a matriz de `response`, pois `response` entrará no ciclo a seguir.
7. Entra em um ciclo `for` com contador k de 1 até o comprimento de `response`.
    1. Cria `response` como a coluna k de `response.all`.
    2. Atribui log(`response`) a `response`.
    3. Atribui log(`reference`) a `reference2`, para não sobrepor o `reference` criado antes do ciclo.
    4. Cria o objeto `response.m` com o modelo linear de `response` em função de `reference2`.
    5. Cria o objeto `f` com os valores da estatística *F* de `response.m`.
    6. Cria o objeto p com o valor de *p* associado ao valor de *F* em `f`.
    7. Se p > `alpha`, escreve "lm not significant" em `result[k]`.
    8. Se a condição acima não for atendida, segue normalmente como abaixo.
    9. Cria o objeto `slope` com a inclinação da reta em `response.m`.
    10. Cria o objeto `slope.se` com o erro padrão da inclinação da reta em `response.m`.
    11. Cria o objeto `tvalue` com o valor absoluto do *t-test* para checar se `slope` é significativamente diferente de `expected.slope`.
    12. Cria o objeto `pvalue` com o valor de *p* referente ao `tvalue` acima.
    13. Se `pvalue` > `alpha`, escreve "Isometry" em `result[k]` e faz a correção isométrica como descrito na contextualização, e a coloca em `result2[,k+1]`.
    14. Se a condição acima não for atendida, escreve "Allometry" em `result[k]` e faz a correção alométrica como descrito na contextualização, e a coloca em `result2[,k+1]`.
    15. Para terminar o ciclo, atribui o nome da coluna k em `response.all` à coluna k de `response`.
8. A primeira coluna de `result2` passa a ser a média de `reference`, que é a correção feita para a variável de referência.
9. Atribui o nome da variável referência e da(s) variável(is) resposta ao nome das colunas de `result2`.
10. Atribui `specimen.id` aos nomes das linhas de `result2`.
11. Cria o objeto `all`, que é um array com uma lista, a qual contém `result` e `result2`.

Saída:

- Um array com dois itens:
    - A primeira lista contém um *data frame* com as variáveis resposta nas colunas, e a única linha indica se há uma relação de isometria ou alometria com a variável de referência;
    - A segunda lista contém uma matriz com os dados transformados de acordo com a relação determinada no passo anterior. Dessa forma, contém as variáveis transformadas nas colunas (referência e resposta(s)) e os espécimes nas linhas.

Plano B: Composição nucleotídica de sequenciamentos de nova geração

Contextualização

Com o advento de novas tecnologias e o crescimento exponencial na informação gerada por sequenciamentos, torna-se necessário um esforço tanto computacional quanto de programação para

melhorar a eficiência da análise destes dados. Uma das informações importantes para serem extraídas de sequenciamentos, principalmente de sequenciamentos de nova geração (NGS), que muitas vezes involvem dados não só da estrutura do genoma como de expressão gênica, é a composição nucleotídica dos fragmentos obtidos. A proporção de A, T, G e C, assim como de AT em relação a GC, pode trazer informações importantes sobre o papel de alguns genes na determinação do viés de códons, sobre a eficiência na tradução de sequências codificantes (Halder et al., 2017), assim como outras informações sobre eficiência e calibragem dos próprios sequenciamentos (Dohm et al., 2008).

Planejamento da função

Entrada: nuc.prop (`filename`, `identifier`)

- `filename` = nome do arquivo de entrada que se encontra no diretório de trabalho (classe: character).
- `identifier` = nome do identificador dos fragmentos sequenciados, que fica logo após o "@" no início da linha de cada fragmento (classe: character).

Verificando os parâmetros:

- `filename` está presente, é um caráter e existe no diretório de trabalho? Se não, para e escreve: "filename should be a character and present in the current working directory".
- `identifier` está presente e é um caráter? Se não, para e escreve: "identifier should be a character that specifies the few letters and or numbers after the @, common to all fragments' names".

Pseudo-código:

1. Cria o objeto `seqfile`, que contém a leitura de linhas do arquivo texto de sequências `filename`.
2. Cria o objeto `start`, que inclui o `identifier`.
3. Cria o objeto `lines`, que faz a busca de padrões pelo `grep`, identificando o número das linhas que contém as sequências, as quais vem logo após a linha com o `identifier`.
4. Cria o objeto `seqs` com `lines` NAs.
5. Entra em um ciclo `for` com o contador `i` de 1 até o comprimento de `lines`.
    1. Retira a linha `i` de `seqfile`, identificada por `lines` como as sequências de interesse, e atribui ao elemento `i` de `seqs`.
6. Cria o objeto `unique`, que conterá as sequências não-ambíguas de `seqs`.
7. Cria os objetos `ratios.A`, `ratios.T`, `ratios.G` e `ratios.C`, `unique` NAs.
8. Entra em um ciclo `for` com o contador `k` de 2 até o comprimento de `unique` (começa de dois pois a primeira sequência de `unique` sempre é composta exclusivamente por Ns).
    1. Cria o objeto `seqsplit`, que retorna a sequência `k` de `unique` com cada letra separada como um caráter.
    2. Utiliza o `grep` para identificar as posições em `seqsplit` em que se encontra um determinado nucleotídeo, e atribui a proporção deste nucleotídeo no fragmento a `ratios.X[k]` (X equivale a um determinado nucleotídeo).
9. Inicia um dispositivo gráfico através de `x11()` e estabelece `par(mfrow=c(2,2))`.
10. Para cada nucleotídeo, constroe o histograma de `ratios.X` com `freq=F` e `ylim=c(0,10)`.
11. Para cada histograma, coloca a linha vertical da média esperada em vermelho, e a linha vertical

da média observada em azul, com `na.rm=TRUE` para o cálculo de `mean`.
12. Retorna `par` para o *default* (`par(mfrow=c(1,1))`).
13. Cria o objeto `base.summary`, que consiste em uma lista com `summary(ratios.X)` para cada base, e também a proporção combinada AT e GC, somando a média de cada um para a combinação, usando `na.rm=TRUE` para o cálculo de `mean`.

Saída:

- Quatro histogramas na mesma tela que contém a distribuição das frequências de cada uma das bases nucleotídicas (A, T, C e G) nos fragmentos. Para cada histograma, haverá uma linha vertical vermelha com a média esperada (distribuição homogênea das bases = 0.25 cada) e uma linha azul com a média encontrada.
- Uma lista dos `summary` das frequências de cada base no sequenciamento e também com a proporção combinada de AT e GC.

Referências

Colton, T. F. (1999). Size and Shape in Biology. Pages 1-44, in Tested studies for laboratory teaching, Volume 20 (S. J. Karcher, Editor). Proceedings of the 20th Workshop/Conference of the Association for Biology Laboratory Education (ABLE), 399 pages.

Dohm, J. C., Lottaz, C., Borodina, T., & Himmelbauer, H. (2008). Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. Nucleic Acids Research, 36(16), e105.

Halder, B., Malakar, A. K., & Chakraborty, S. (2017). Nucleotide composition determines the role of translational efficiency in human genes. Bioinformation, 13(2), 46–53.

Lleonart J., Salat J., Torres G.J. (2000). Removing allometric effects of body size in morphological analysis. Journal of Theoretical Biology, 205, 85-93.

Comentários Danilo

Jonathan, seus pseudocódigos estão bem claros, mas, para mim, suas propostas parecem muito simples.

Se eu entendi bem, seu plano A basicamente roda um modelo linear e re-organiza os resultados em um formato útil para quem trabalha com morfometria. E o plano B conta ocorrências de bases e pares de bases, de forma que ficou parecendo uma espécie de table() vitaminado.

O ideal, na minha avaliação, seria você bolar uma terceira proposta (um plano C) que exija um algoritmo um pouco mais complexo. Se vc for bolar um plano C, eu sugiro incluir depois da contextualização uma frase resumindo o que a função faz (antes da especificação dos parâmetros).

Mas, se você não tiver nenhuma ideia, eu sugiro fazer o plano B, pois aparentemente ele pode ser um desafio mais interessante.

Danilo

Links para o trabalho final

Para o trabalho final, resolvi seguir com o Plano A, pois achei que seria mais útil e mais desafiador, principalmente pelos vários fatores que encontrei que podem complicar a leitura de um data frame e das variáveis, da forma que o usuário deseja transformá-las. Como mencionei, a função checa se a relação entre cada variável resposta e a variável referência é isométrica ou alométrica, e faz a transformação destas variáveis de acordo com a relação, gerando uma tabela com os dados transformados. Nessa nova tabela, as diferenças de tamanho entre os espécimes analisados foram eliminadas, e portanto agora são comparáveis e podem ser submetidos a outras análises.

* Link para a página da função: Isometry test and size correction

* Link para a página de ajuda da função: Help