

# Marianne Azevedo Silva



Sou Graduada em Ciências Biológicas pela UNICAMP e mestre pelo Programa de Pós Graduação em Ecologia da UNICAMP sob orientação do Prof. Paulo S. Oliveira. Atualmente sou doutoranda pelo mesmo Programa e orientador. Tenho minha formação voltada a ecologia molecular. Em minhas pesquisas tenho especial interesse em descrever os padrões locais de distribuição da diversidade genética intraespecífica, bem como entender os fatores que a determinam e a mantém em sistemas naturais. Atualmente, o sistema biológico com a qual trabalho inclui formigas do Cerrado, embora também tenha interesse em outros sistemas.

---

---

## I. Meus Exercícios

Link para a página com os meus exercícios resolvidos [Exercícios](#)

---

---

## II. Trabalho Final - Propostas

### II.I Proposta A: Suficiência amostral em trabalhos com diversidade genética

#### Contextualização

Tamanho amostral adequado é essencial em qualquer estudo empírico cujo objetivo seja fazer inferências sobre a população a partir de uma amostragem<sup>1)</sup>. Em geral, avaliar a suficiência amostral é uma prática frequente em trabalhos de ecologia. No entanto, em outras áreas, como genética de populações, tal abordagem ainda é dificilmente incorporada às análises de dados. Alguns trabalhos tem buscado estabelecer generalizações a respeito do tamanho amostral adequado para estimativa de diversidade genética das populações (e.g. Hale *et al.* 2012). Apesar desse esforço, é sabido que essa suficiência amostral pode variar com o tipo de marcador molecular, nível de polimorfismo dos *loci* e história natural das espécies. Sendo assim, esta proposta tem como objetivo desenvolver uma função que permita aos pesquisadores interessados em genética de populações avaliar se, dado o conjunto de *loci* microsatélites genotipado, seu esforço amostral foi suficiente para inferir a diversidade genética das populações ou grupos de indivíduos diplóides sob estudo.

#### Planejamento da Função

**Entrada:** sample.suff (dados, col.ID, col.pop, col.loci, na.code, IC.plot, gen.div, nsim)

- dados = data frame no qual cada linha corresponde a um indivíduo e as colunas possuem informação da identificação do indivíduo, identificação da população e alelos dos *loci* genotipados
- col.ID = número da coluna com identificação dos indivíduos (classe: integer, col.ID>0)
- col.pop = número da coluna com identificação da população a qual o indivíduo pertence (classe: integer, col.pop>0)
- col.loci = intervalo que corresponde às colunas correspondentes aos *loci* (classe: integer, col.loci>0)
- na.code = código utilizado para dados faltantes
- IC.plot = vetor lógico (TRUE/FALSE) indicando se intervalos de confiança devem ser plotados
- gen.div = estimativa de diversidade genética a ser calculada, podendo ser: Heterozigosidade esperada e observada (He e Ho, respectivamente), riqueza alélica (Ar) ou riqueza de alelos privados (pA) (classe: character)
- nsim = número de simulações (classe: integer, nsim>0)
- Observação: Caso haja tempo disponível, pretende-se colocar na função o argumento mol.marker, o qual permitirá escolher entre marcadores microssatélites e marcadores SNPs

## Verificando os parâmetros

- dados é da classe data.frame com 1 indivíduo por linha? Se não, pára e escreve: “Objeto de entrada deve ser um data.frame com um indivíduo por linha”
- col.ID, col.pop e col.loci são números inteiros e maiores que zero? Se não, pára e escreve: “col.ID, col.pop e col.loci devem ser números inteiros maiores que zero”
- cada *locus* está organizado em duas colunas, uma para cada alelo? Se não, pára e escreve “Cada *locus* deve estar organizado em duas colunas correspondentes aos alelos. Por exemplo (colocar uma linha de exemplo)”
- IC.plot é um vetor lógico? Se não, pára e escreve: “IC.plot deve ser um vetor lógico (TRUE/FALSE)”
- gen.div é um objeto da classe character com uma das opções de estimativa de diversidade genética (He, Ho, Ar, pA)? Se não, pára e escreve: “gen.div deve ser uma das opções: “He”, “Ho”, “Ar”, “pA” ”
- nasim é um número inteiro maior que zero? Se não, pára e escreve: “nasim deve ser um número inteiro maior que zero”
  - Se nasim for menor que 1000, continua, mas dá um warning: “Um número baixo de simulações pode afetar seus resultados!”

## Pseudo-código

1. Carregar pacotes necessários (PopGenReport<sup>2)</sup> e Adegenet<sup>3)</sup>)
2. Transformar dados (data frame) em objeto genind (para isso, os argumentos col.ID, col.pop e col.loci serão usados)
3. Criar um objeto simula.allpop como sendo uma lista vazia (permite inserção de quantos objetos forem necessários dentro da lista)
4. Entrar em um ciclo (1) for com contador i de 1 até o número total de populações. Ou seja, para cada população irá fazer:
  1. Criar objeto simula.media com NAs correspondente ao número de indivíduos da população
  2. Criar objeto simula.ICsup com NAs correspondente ao número de indivíduos da população
  3. Criar objeto simula.ICinf com NAs correspondente ao número de indivíduos da

população

4. Entrar em um ciclo (2) for com contador j de 1 até número de indivíduos da população
  1. Criar objeto `simula.nind` com NAs correspondente a `nsim`
  2. Entrar em um ciclo(3) 'for' com contador k de 1 até `nsim`
    1. simular a estimativa escolhida de `gen.div` para uma subamostragem de tamanho j da população (usar função `sample` com `size=j` e `replace=T`). Cada simulação é armazenada no objetos `simula.nind` na posição k
    3. Armazenar a média de `simula.nind` no objeto `simula.media` na posição j
    4. Armazenar o intervalo de confiança superior de `simula.nind` no objeto `simula.ICsup` na posição j
    5. Armazenar o intervalo de confiança inferior de `simula.nind` no objeto `simula.ICinf` na posição j
5. Armazenar no objeto `simula.allpop` na posição i um data frame com 5 colunas sendo elas: numero de individuos amostrados (de 1 até número total de indivíduos na população), populacao (nome da população repetido em todas as linhas), media (médias armazenadas no objeto `simula.media`), ICsup (intervalos de confiança superiores armazenados no objeto `simula.ICsup`), ICinf (ntervalos de confiança inferiores armazenados no objeto `simula.ICinf`)
5. Juntar todos os data frames do objeto `simula.allpop` em um único data frame `gen.div.rarefaction`
6. Plotar gráfico da estimativa de diversidade genética em função do número de indivíduos. Para o plot usar: (i) função `layout()` - no lado esquerdo o gráfico e no direito legenda com as cores de cada população; (ii) uma cor para cada população
7. Se `IC.plot` for igual a TRUE, plotar no gráfico as linhas correspondentes aos intervalos de confiança superiores e inferiores das simulações. Nesse caso, as linhas devem ser da cor da população correspondente, mas com um tipo diferente. Legenda deve conter as cores das populações e o que cada tipo de linha indica.
8. Retornar uma lista sendo que o primeiro objeto é a estimativa de diversidade genética observada para as populações e o segundo objeto é o data frame `gen.div.rarefaction`

## Saída

- Gráfico com estimativa de diversidade genética em função do tamanho amostral simulado da população (podendo ou não apresentar o intervalo de confiança das simulações)
- Objeto da classe list (`gen.div.rarefaction`), contendo, para cada população, a média e os intervalos de confiança superior e inferior da estimativa de diversidade genética em cenários simulados com diferentes números de indivíduos.

Comentários Vitor Rios

Mariana, parabéns pelas propostas, bem pensadas e bem estruturadas. Você pode seguir com qualquer uma das duas, mas eu gostei mais da primeira, me parece que seria útil para quem trabalha com genética de populações [Vitor](#)

Resposta aos Comentários

Oi Vitor! Obrigada por ler as propostas e pelo feedback! Vou seguir com a primeira proposta. Em caso de dúvidas, escrevo para você 😊

## II.II Proposta B: Adequação alimentar

### Contextualização

A população brasileira tem sido, com alta frequência, alvo de doenças como desnutrição, obesidade, diabetes e hipertensão, todas amplamente associadas a hábitos alimentares inadequados. Frente a isso, esforços governamentais tem sido aplicados a fim de orientar os brasileiros a terem uma alimentação mais saudável e com escolhas alimentares adequadas. Para atingir esse objetivo, o Ministério da Saúde lançou o “Guia alimentar para uma alimentação saudável”<sup>4)5)</sup>, o qual é atualizado periodicamente. Com o intuito de auxiliar a adequação alimentar de brasileiros adultos, essa proposta tem a finalidade de desenvolver uma função que avalie se uma pessoa:

1. está dentro do limite de peso ideal para seu sexo e altura
2. está tendo uma ingestão adequada de macronutrientes segundo as diretrizes da Organização Mundial da Saúde (diretrizes estas também presentes no guia alimentar do Ministério da Saúde)

Caso o peso e alimentação não estejam de acordo com as estimativas ideais, e se for do desejo da usuário, a função pode retornar um objeto com sugestões de consumo diário variado para uma alimentação saudável que esteja de acordo com as orientações do Ministério da Saúde.

### Planejamento da função

**Entrada:** adeq.alimentar (recordatorio.semanal, peso, altura, idade, sexo, ativ.fisica, cardapio)

- recordatorio.semanal = data frame com o recordatório de ao menos uma semana (7 dias) da alimentação do usuário. Cada linha deve corresponder a um dia da semana e deve haver 5 colunas, sendo elas: n.refeicoes (número de refeições feitas no dia), P (quantidade de proteínas ingeridas no dia), L (quantidade de lipídios ingeridos no dia), C (quantidade de carboidratos complexos ingeridos no dia) e A (quantidade de açúcar livre consumido no dia). As colunas de P a A devem estar em unidade de quilocalorias (usuário pode ser basear no Anexo C [deste guia](#) para montar seu data frame).
- peso = massa do usuário em quilogramas (classe: numeric, peso>0)
- altura = altura do usuário em metros (classe: numeric, altura>0)
- idade = idade do usuário em anos (classe: integer, 19<idade<65)
- sexo = sexo do usuário (classe: character, devendo ser “H” para homens ou “M” para mulheres)
- ativ.fisica = intensidade da atividade física realizada pelo usuário (classe: character, podendo ser “sedentario”, “pouco ativo”, “ativo” ou “muito ativo”)
- cardapio = vetor lógico (TRUE/FALSE) indicando se sugestões de consumo diário variado para adequação alimentar devem ser geradas

### Verificando parâmetros

- recordatorio.semanal é um data frame, contendo, pelo menos, 7 linhas e 5 colunas, tal que cada linha corresponde a um dia da semana e as colunas correspondem a n.refeicoes, P, L, C e A? se não, pára e escreve: “Recordatório deve ser um data frame, onde cada linha corresponde a um dia da semana e as colunas devem seguir o seguinte formato: (colocar uma linha com um

exemplo)”

- peso e altura são números maiores que zero? Se não, pára e escreve: “peso e altura devem ser números maiores que zero”
- idade é um número inteiro maior que 19 e menor que 65? Se não, aparecer um warning: “Cálculos incluídos nesta função foram desenvolvidos para indivíduos adultos. Para crianças, adolescentes e idosos há cálculos específicos que não estão incluídos nessa função!”
- sexo é igual a “H” ou “M”? Se não, pára e escreve: “sexo deve ser H (para homens) ou M (para mulheres)”
- `ativ.fisica` é um objeto da classe `character` com uma das opções de intensidade da atividade física (sedentário, pouco ativo, ativo ou muito ativo? Se não, pára e escreve: “`ativ.fisica` deve ser uma das opções: “sedentário”, “pouco ativo”, “ativo” ou “muito ativo” ”
- `cardapio` é um vetor lógico? Se não, pára e escreve: “`cardapio` deve ser um vetor lógico (TRUE/FALSE)”

## Pseudo-código

1. Estimar peso ideal  $\pi$ <sup>6)</sup>
  1. se `sexo` = “H”, IMC desejável = 22 Kg/m<sup>2</sup>
  2. se `sexo` = “M”, IMC desejável = 21 Kg/m<sup>2</sup>
2. Estimar adequação de peso  $ap \Rightarrow ap = (\text{peso}/\pi)*100$
3. Plotar peso e  $\pi$  com intervalos de confiança de mais ou menos 10% de  $\pi$ <sup>7)</sup>
  1. se  $ap \leq 70$ , colocar um texto no gráfico “Desnutrição grave”
  2. se  $70.1 \leq ap \leq 80$ , colocar um texto no gráfico “Desnutrição moderada”
  3. se  $80.1 \leq ap \leq 90$ , colocar um texto no gráfico “Desnutrição leve”
  4. se  $90.1 \leq ap \leq 110$ , colocar um texto no gráfico “Eutrofia (dentro do peso ideal)”
  5. se  $110.1 \leq ap \leq 120$ , colocar um texto no gráfico “Sobrepeso”
  6. se  $ap > 120$ , colocar um texto no gráfico “Obesidade”
4. Substituir a string de `ativ.fisica` por valor correspondente
  1. se `sexo` = “H”:
    1. se `ativ.fisica` = “sedentário” substitui pelo valor 1
    2. se `ativ.fisica` = “pouco ativo” substitui pelo valor 1.11
    3. se `ativ.fisica` = “ativo” substitui pelo valor 1.25
    4. se `ativ.fisica` = “muito ativo” substitui pelo valor 1.48
  2. se `sexo` = “M”:
    1. se `ativ.fisica` = “sedentário” substitui pelo valor 1
    2. se `ativ.fisica` = “pouco ativo” substitui pelo valor 1.12
    3. se `ativ.fisica` = “ativo” substitui pelo valor 1.27
    4. se `ativ.fisica` = “muito ativo” substitui pelo valor 1.45
5. Calcular necessidade de energia estimada  $nee$  para o peso
  1. se `sexo` = “H” usar a fórmula:  $662 - (9.53 * idade) + \text{ativ.fisica} * (15.9 * \text{peso}) + (539.6 * \text{altura})$
  2. se `sexo` = “M” usar a fórmula:  $354 - (6.91 * idade) + \text{ativ.fisica} * (9.36 * \text{peso}) + (726 * \text{altura})$
6. Calcular média de `recordatorio.semanal$n.refeicoes` e armazenar no objeto `mean.n.refeicoes`
7. Fazer um teste t para ver se `mean.n.refeicoes` é significativamente diferente de 5<sup>8)</sup>
  1. se teste t for significativo, escrever na tela “Você deve ajustar seu número de refeições para 5 refeições ao dia, incluindo café da manhã, lanche da manhã, almoço, lanche da tarde e janta”
8. Calcular intervalos de confiança superior e inferior da média das porcentagens diárias de `recordatorio.semanal$P` em relação a  $nee$  e armazenar nos objetos `ICsup.p` e `ICinf.p`,

respectivamente

1. se  $IC_{sup.p}$  e  $IC_{inf.p}$  estiverem contidos no intervalo de 15% a 20%<sup>9)</sup>, escrever “Seu consumo diário de proteínas está bom!”
  2. caso contrário, escrever “Seu consumo diário de proteínas precisa ser ajustado”
9. Calcular intervalos de confiança superior e inferior da média das porcentagens diárias de `recordatorio.semanal$L` em relação a `nee` e armazenar nos objetos  $IC_{sup.l}$  e  $IC_{inf.l}$ , respectivamente
1. se  $IC_{sup.l}$  e  $IC_{inf.l}$  estiverem contidos no intervalo de 20% a 30%<sup>10)</sup>, escrever “Seu consumo diário de lipídios está bom!”
  2. caso contrário, escrever “Seu consumo diário de lipídios precisa ser ajustado”
10. Calcular intervalos de confiança superior e inferior da média das porcentagens diárias de `recordatorio.semanal$C` em relação a `nee` e armazenar nos objetos  $IC_{sup.c}$  e  $IC_{inf.c}$ , respectivamente
1. se  $IC_{sup.c}$  e  $IC_{inf.c}$  estiverem contidos no intervalo de 45% a 65%<sup>11)</sup>, escrever “Seu consumo diário de carboidratos complexos está bom!”
  2. caso contrário, escrever “Seu consumo diário de carboidratos complexos precisa ser ajustado”
11. Calcular intervalos de confiança superior e inferior da média das porcentagens diárias de `recordatorio.semanal$A` em relação a `nee` e armazenar nos objetos  $IC_{sup.a}$  e  $IC_{inf.a}$ , respectivamente
1. se  $IC_{sup.a}$  e  $IC_{inf.a}$  estiverem contidos no intervalo de 0% a 10%<sup>12)</sup>, escrever “Seu consumo diário de açúcares simples está bom!”
  2. caso contrário, escrever “Seu consumo diário de açúcares simples precisa ser ajustado”
12. Se `cardapio = TRUE`
1. Carregar uma base de dados de alimentos disponibilizado no Anexo C [deste guia](#)<sup>13)</sup>
  2. Calcular necessidade de energia estimada ideal `nee.ideal` para o `pi`
    1. se `sexo = "H"` usar a fórmula:  $662 - (9.53 * idade) + ativ.fisica * (15.9 * pi) + (539.6 * altura)$
    2. se `sexo = "M"` usar a fórmula:  $354 - (6.91 * idade) + ativ.fisica * (9.36 * pi) + (726 * altura)$
  3. criar um objeto `sugestoes.diarias` da classe `lista.vazio`
  4. se  $90.1 \leq ap \leq 110$ , entrar em um ciclo `for` com contador `i` de 1 até 7
    1. para cada `i` amostrar alimentos (com reposição) tal que a soma total de kcal seja igual a `nee` e cuja partição respeite as porcentagens de cada grupo de alimentos sugeridas pelo Ministério da saúde<sup>14)</sup>. Armazenar cada ciclo em `sugestoes.diarias[i]`
  5. se  $ap < 90$ , entrar em um ciclo `for` com contador `i` de 1 até 7
    1. para cada `i` amostrar alimentos (com reposição) tal que a soma total de kcal seja igual a `nee.ideal + 500 cal` e cuja partição respeite as porcentagens de cada grupo de alimentos sugeridas pelo Ministério da saúde. Armazenar cada ciclo em `sugestoes.diarias[i]`
  6. se  $ap > 110$ , entrar em um ciclo `for` com contador `i` de 1 até 7
    1. para cada `i` amostrar alimentos (com reposição) tal que a soma total de kcal seja igual a `nee.ideal - 500 cal` e cuja partição respeite as porcentagens de cada grupo de alimentos sugeridas pelo Ministério da saúde. Armazenar cada ciclo em `sugestoes.diarias[i]`
13. Retornar o objeto `sugestoes.diarias`

## Saída

- Gráfico comparativo de peso atual do usuário com o peso ideal, com informação do nível de nutrição do usuário
- Impressões na tela com relação à adequação do consumo de macronutrientes do usuário em relação às diretrizes da OMS
- Objeto da classe `list(sugestoes.diarias)`, contendo, para cada dia da semana, uma lista de sugestões de alimentos adequados para manutenção, ganho ou perda de peso do usuário e que respeita a proporção diária de cada grupo de alimento sugerida pelo Ministério da Saúde.

**Observações adicionais:** Dadas as condições que devem ser respeitadas em cada ciclo for da proposta B e minhas limitações como usuária do R, para desenvolver tais ciclos provavelmente será necessário ajuda dos monitores. Por isso, os ciclos não estão bem desenvolvidos nesta proposta.

---

---

## III. Trabalho Final - Versão Final

### III.I Proposta escolhida

Link para a página com a proposta escolhida, incluindo alterações e respectivas justificativas: [Proposta escolhida](#)

### III.II Código da função e arquivo da função

Link para a página com o código e o arquivo da função: [Função sample.suff](#)

### III.III Página de ajuda (help) da função

Link para a página com o help e o arquivo contendo o conjunto de dados de exemplo para executar a função: [Página de ajuda \(help\) da função sample.suff](#)

- 1) Fonte: [https://en.wikipedia.org/wiki/Sample\\_size\\_determination](https://en.wikipedia.org/wiki/Sample_size_determination)
- 2) <https://cran.r-project.org/web/packages/PopGenReport/index.html>
- 3) <https://cran.r-project.org/web/packages/adegenet/index.html>
- 4) [http://bvsmms.saude.gov.br/bvs/publicacoes/guia\\_alimentar\\_populacao\\_brasileira\\_2008.pdf](http://bvsmms.saude.gov.br/bvs/publicacoes/guia_alimentar_populacao_brasileira_2008.pdf)
- 5) [http://189.28.128.100/dab/docs/portaldab/publicacoes/guia\\_alimentar\\_bolso.pdf](http://189.28.128.100/dab/docs/portaldab/publicacoes/guia_alimentar_bolso.pdf)
- 6)  $\pi = \text{IMC desejável para adultos} * \text{altura}^2$
- 7) variação de  $\pi$  que ainda é considerada como peso ideal
- 8) número ideal de refeições ao dia (café, lanche da manhã, almoço, lanche da tarde e janta)
- 9)

consumo diário de proteínas em relação ao NEE segundo a OMS

10)

consumo diário de lipídios em relação ao NEE segundo a OMS

11)

consumo diário de carboidratos complexos em relação ao NEE segundo a OMS

12)

consumo diário de açúcares simples em relação ao NEE segundo a OMS

13)

será reorganizada em um único data frame que inclua uma coluna \$tipo correspondente ao grupo de alimentos

14)

45% cereais/tubérculos/raízes; 3% feijões; 11% frutas; 3% legumes e verduras; 18% leite e derivados; 10% carnes e ovos; 4% óleo e gordura; 6% açúcares simples

From:

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

Permanent link:

[http://ecor.ib.usp.br/doku.php?id=05\\_curso\\_antigo:r2018:alunos:trabalho\\_final:azevedosilva.m:start](http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2018:alunos:trabalho_final:azevedosilva.m:start) 

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