Louise Alissa de Morais

Mestranda no Departamento de Ecologia da Universidade de São Paulo, orientada por Glauco Machado.

Interessada em comportamento sexual e seleção sexual. Tentando fazer uma transição pouco dolorosa dos trabalhos de campo para as estatística e modelagens.

Exercícios

Você poderá visualizar meus exercícios no seguinte link → execícios Louise Alissa

---

Propostas de Trabalho

Veja nesse link minhas propostas iniciais e os comentários feito pelas monitoras → Propostas A e B

> Monitoras queridas,
>
> Fiquei pesquisando como inserir o *m* (de seleção de parceiros) na proposta e parece que usá-lo da mesma forma que se usa o *s* é bastante ingênuo e simplista. Poderia ser usado de forma didática, mas parece que passa do limite da simplificação e chega a ser errado. É por esse motivo que não estudamos dessa forma na disciplina de “Processos Evolutivos”.
>
> Quebrei a cabeça, pedi ajuda para pessoas mais instruídas e elas me disseram que fazer deriva + seleção já é trabalho suficiente, já que trabalharei com várias matrizes e *for*s. Vou tentar fazer uns detalhes bonitinhos pra ficar uma função legal. Se eu ver vocês posso explicar direitinho os problemas conceituais.

PROPOSTA FINAL

Função genpop()

Meu primeiro contato com o R foi com um script na disciplina “Processos Evolutivos”, e me lembro de

ter ficado um pouco impressionada com a linguagem. Tive a sensação que era complicado demais e que nunca iria conseguir programar. Nessa disciplina, parte dos estudos sobre Equilíbrio de Hard-Weinberg era feito em um programa chamado *Populus* e uma outra parte por alguns scripts. Fazendo com que o conteúdo do script fique dentro de uma função, a atividade pode ser mais palatável e interativa para alguns alunos.

Input e output

Minha proposta é fazer uma função que retorne graficamente a frequência de alelos ao longo do tempo, segundo a Teoria de Hard- Weinberg. O gráfico apresentará as flutuações dos valores do alelo p, em uma ou mais populações, ao longo das gerações.

Quero que minha função funcione em um cenário com a ausência de seleção natural, no qual poderemos estudar deriva, e na presença de seleção natural. Para isso, seria necessário que a função contasse com, pelo menos, os seguintes argumentos:

- n = , referente ao tamanho das populações (número de indivíduos)
- npop = , referente ao número de populações
- ntime= , referente ao número de gerações
- freqi= , frequência inicial do alelo p (na literatura é padrão usar *p* para “azão”, q para “azinho”, sendo que $p + q = 1$)
- sel.on= , argumento que indicará se a seleção recai sobre o fenótipo dominante ou recessivo
- s= 0, default é zero, o que significa que não há seleção (varia de 0 a 1)

Minha função é para apenas 2 alelos (*p* e *q*) e considera que o tamanho da população (n) é igual ao número efetivo e que os indivíduos se reproduzem de forma pan-mítica. Além disso, a população inicial está em equilíbrio de H-W, que poderá ser posteriormente perturbado se houver seleção.

Entrega da Proposta

FUNÇÃO genpop

Veja abaixo o help da minha função ou baixe o arquivo → help.txt

```
genpop{}                        Package: unknown                        R
Documentation


Função didática para estudo do Princípio de HArdy-Weinberg


Description:

genpop simula e grafica a frequência do alelo p ao longo de gerações em
diferentes
```

```
cenários. Se s=0, a flutuaçao nas frequêcias se dará apenas por deriva
genética
(processo estocástico), mas, se s>0, as frequências flutuarão devido à
deriva genética
e seleção natural (processo de terminístico).


Usage:

genpop(n, npop, freqi, ntime, s=0, sel.dom="no")


Arguments:

n           numérico; número de populações que serão simuladas.

npop        numérico; número de indivíduos em cada população.

freqi       numérico; frequência inicial do alelo p nas populações. Valor
deve
            estar entre 0 e 1.

ntime       numérico; número de gerações simuladas.

s           numérico; coeficiente de seleção natural. Valor deve estar entre
zero
            e 1, sendo que, quanto maior o valor, mais intensa será a
seleção. O
            valor default é 0.

sel.dom     lógico; se for TRUE a seleção recairá sobre os fenótipos
dominantes,
            se for FALSE a seleção recairá sobre o fenótipo recessivo. O
valor
            defaut é "no", apenas para situações em que s=0.


Details:

A função genpop baseia-se nas premissas de que o gene em questão possui
apenas
dois alelos, p e q, e se expressa por dominância completa. Por ser bi-
alélico,
isso significa que ao inserir freqi=0.2, teremos p=0.2 e q=0.8. E relação à
dominância completa temos os genótipos (p^2) e (2*p*q) que expressam o
fenótipo
dominante e o genótipo (q^2) que expressa o fenótipo recessivo. Além disso,
outras premissas são que o número de indivíduos na população é igual ao
tamanho
efetivo populacional e os organismos se reproduzem de forma pan-mítica.
```

Value:

A função genpop retorna um gráfico com a frequência do alelo p no eixo y e o número de gerações no eixo x.
O ponto em preto marca o valor inicial de p e a linha preta tracejada é a trajetória determinística (sem deriva) esperada.
Cada uma das linhas coloridas representa a flutuação na frequência do alelo p em uma população.
Na parte superior da janela gráfica está reportado os valores que foram inseridos
nos argumentos, e também o número de populações em que alelos foram fixados.
Populações com valores de p=1 fixaram o alelo p, enquanto que populações com valores de p=0 fixaram o alelo q.

Warnings:

Caso seja escolhido valores muito grandes para os argumentos n, npop e ntime, a
função pode demorar para retornar o gráfico. Se for simular a trajetória da frequência do alelo por muitas gerações, esteja ciente de que os alelos podem
ser fixados em períodos curtos de tempo. Se isso acontecer, as linhas ficarão
aglutinadas no início do gráfico.

Notes:

Ao selecionar um valor para o coeficiente de seleção maior que zero, s>0, é necessário indicar o fenótipo que sofrerá a selação no argumento sel.dom.
Caso contrário, se s=0, o argumento sel.dom estar com o valor default.

Author(s):

Função desenvolvida por Louise M. Alissa (2016).
louisem.alissa@gmail.com

References:

Ridley, M. (2006) Evolução. Artmed, 3ªed.

See also:

package:HardyWeinberg.
Enquanto a função genpop é apenas uma simplificação para fins didáticos, esse

```
pacote possui testes estatísticos e gráficos para análise de equilíbrio de
Hardy
-Weinberg.


Examples:

## Vendo efeito de deriva:
# populações pequenas fixam alelos em menor número de gerações
genpop(n=20, npop=200, freqi=0.5, ntime=300, s=0, sel.dom="no")
genpop(n=20, npop=20, freqi=0.5, ntime=300) # não precisa especificar se s=0

## Vendo efeito se seleção no dominante:
# p selecionado negativamente; veja as diferentes intensidades
genpop(n=20, npop=100, freqi=0.5, ntime=100, s=0.1, sel.dom=TRUE)
genpop(n=20, npop=100, freqi=0.5, ntime=100, s=0.2, sel.dom=TRUE)

## Vendo efeito de seleção no recessivo:
# p selecionado positivamente; veja as diferentes intensidades
genpop(n=20, npop=100, freqi=0.5, ntime=100, s=0.1, sel.dom=FALSE)
genpop(n=20, npop=100, freqi=0.5, ntime=100, s=0.2, sel.dom=FALSE)

## Vendo efeito do tempo:
# dado o tempo necessário há fixação
genpop(n=20, npop=200, freqi=0.5, ntime=100, s=0, sel.dom="no")
genpop(n=20, npop=200, freqi=0.5, ntime=1000, s=0, sel.dom="no")
```

Veja abaixo o código da minha função, ou baixe o arquivo → genpop.r

```
genpop <- function(n, npop, freqi, ntime, s=0, sel.dom="no") # Nomeando a
função e os argumentos, apenas s e sel.dom possuem valores default
{
  ### Avisos para possíveis erros na entrada dos argumentos ###
  cat("ATENÇÃO: Caso tenha escolhido valores altos para n, npop e ntime, a
função pode demorar\n para retornar o gráfico.\n") # Para não se assustarem
com demoras.
  if(n<=0 | npop<=0 | ntime<=0) # Se os argumentos forem menores ou iguais a
zero:
  {
    stop("ATENÇÃO: Os argumentos n, npop e ntime devem possuir valores
maiores que 0.") # Avise e pare a função, pois esses argumentos devem ser
corretamente preenchidos.
  }
  if(freqi<0 | freqi>1) # Se o argumento freqi, que é o valor de p inicial,
for menor que zero ou maior que um:
  {
    stop("ATENÇÃO: O argumento freqi é uma proporção, seu valor deve estar
entre 0 e 1.") # Avise e pare a função, pois freqi dever ter a entrada
```

```
conforme formato de proporção.
  }
  if(s>0 & sel.dom=="no") # Se houver designado um valor de s sem alterar o
default do argumento sel.dom:
  {
    stop("ATENÇÃO: Ao selecionar um valor de coeficiente de seleção, s,
informe se este  recairá \nsobre o fenótipo dominante (sel.dom=TRUE) ou
sobre o recessivo (sel.dom=FALSE).") # Avise e para a função para que seja
corrigido a entrada de valores.
  }
  if(sel.dom!="no" & sel.dom!=TRUE & sel.dom!=FALSE)
  {
    stop("ATENÇÃO: O argumento sel.dom deve ser o valor default (se s=0) ou
preenchido com TRUE ou FALSE (se s>0).")
  }
  ### Criando a matriz ###
  matriz.pop <- matrix(NA, nrow=ntime, ncol=n) # Crie uma matriz preenchida
por NA's, com o número de linhas igual ao número de gerações e o número de
colunas igual ao número de populações.
  matriz.pop[1, ]<- freqi # Insira na linha 1, para todas as colunas, o
valor incial de p
  ### Preenchendo a matriz se não houver seleção ###
  if(s==0) # Se s for igual a 0:
  {
    for (colu in 1:n) #Preencha primeiro pelas colunas, todas as n colunas
    {
      for (lin in 2:ntime) # Siga preenchendo as linhas, exceto a primeira
que é a freqi
      {
        p.rep <- matriz.pop[lin-1,colu] # Use a linha anterior pra pegar o
valor de p na geração "passada"
        sample.rep <- sample(c("p","q"), size= npop, replace=T,
prob=c(p.rep, 1-p.rep)) # Para reproduzir a geração "passada", sem seleção,
apenas faça um sample. A reprodução será aleatória, panmítica, mas também
sufeita à variações do sorteio, pois há deriva. Use os valores de proporção
de p e q da geraçao "passada" para constituir a geração seguinte.
        pnovo <- (sum(sample.rep=="p"))/ npop # Do vetor gerado pelo sample,
calcule os valores de p na geração seguinte
        matriz.pop[lin,colu] <- pnovo #Coloque o valor de p da geração
seguinte na célula
      }
    }
  }
  ### Preenchendo a matriz se houver seleção ###
  ## Para calcular os valores de p nas gerações seguintes, com seleção,
utilizasse a formula de Hardy-Weinberg: p^2 + 2*p*q + q^2 = 1. Quando há o
coeficiente de seleção ele é subtraido do fitness do fenótipo: W(fitness) =
1-s(coeficiente de seleção). Para calcular as probabilidades de p e q na
nova geração, devemos multiplicar a probabilidade do fenótipo(AA-p^2 ,
Aa-2*p*q , aa-q^2 ) por 1 (se ele não for afetado por s) ou por 1-s (se a
```

```
seleção recair sobre o fenótipo), e só depois calcular os valores de p e q
novos.
  if(s>0) # Se s for maior que zero, há seleção.
  {
    ## E se a seleção recair sobre o recessivo,(1-s) será aplicado apenas no
q^2(aa):
    if(sel.dom==FALSE) # Se a seleção NÃO for no dominante
    {
      for (colu2 in 1:n) # Preencha primeiro pelas colunas, todas as n
colunas
      {
        for (lin2 in 2:ntime) # Siga preenchendo as linhas, exceto a
primeira que é a freqi
        {
          p.ant <- matriz.pop[lin2-1,colu2] # Use a linha anterior pra pegar
o valor de p na geração "passada"
          q.ant <- 1-p.ant # pois, p + q é igual a 1
          p.rep2 <- (p.ant^2)/(1-((q.ant^2)*s)) + 0.5*((2*p.ant*q.ant)/(1-
((q.ant^2)*s))) # Essa é a formula de H-W, onde multipliquei q^2 por (1-s),
depois normalizei a equação pra que a soma ainda resultasse em 1. Por fim,
isolei a nova probabilidade de p na nova geração
          sample.rep2 <- sample(c("p","q"), size= npop,replace=T,
prob=c(p.rep2, 1-p.rep2)) # Para reproduzir a geração passada com seleção,
use as probabilidades de p calculadas com a aplicação do s, mas também faça
um sample, já que a deriva ainda tem efeito aqui.
          pnovo2 <- (sum(sample.rep2=="p"))/ npop # Calcule a proporção de p
nesse sample
          matriz.pop[lin2,colu2] <- pnovo2 # Coloque o valor de p da geração
seguinte na célula
        }
      }
    }
    ## E se a seleção recair sobre o dominante,(1-s) será aplicado ao
p^2(AA) e 2*p*q(Aa):
    if(sel.dom==TRUE) # Se a seleçao for no dominante
    {
      for (colu3 in 1:n) # Preencha primeiro pelas colunas, todas as n
colunas
      {
        for (lin3 in 2:ntime) # Siga preenchendo as linhas, exceto a
primeira que é a freqi
        {
          p_ant <- matriz.pop[lin3-1,colu3] # Use a linha anterior pra pegar
o valor de p na geração "passada"
          q_ant <- 1-p_ant # pois, p + q é igual a 1
          p.rep3 <- ((p_ant^2)*(1-s)/(1-((p_ant^2)*s)-(2*p_ant*q_ant*s))) +
0.5*((2*p_ant*q_ant*(1-s))/(1-((p_ant^2)*s)-(2*p_ant*q_ant*s))) # Essa é a
formula de H-W, onde multipliquei p^2 e 2*p*q por (1-s), depois normalizei a
equação pra que a soma ainda resultasse em 1. Por fim, isolei a nova
probabilidade de p na nova geração
          sample.rep3 <- sample(c("p","q"), size= npop,replace=T,
```

```
prob=c(p.rep3, 1-p.rep3)) # Para reproduzir a geração passada com seleção, use as probabilidades de p calculadas com a aplicação do s, mas também faça um sample, já que a deriva ainda tem efeito aqui.
          pnovo3 <- sum(sample.rep3=="p") / npop # Calcule a proporção de p nesse sample
          matriz.pop[lin3,colu3] <- pnovo3 # Coloque o valor de p da geração seguinte na célula
        }
      }
    }
  }
  ### Contabilizando o número de populações que fixaram p ou q ###
  fixp <- sum(matriz.pop[ntime, ]==1) # Olhe pra última linha da matriz, o tempo final, e calcule quantas populações chegaram a ter o p=1. Esse é o número das populações que fixaram o alelo p.
  fixq <- sum(matriz.pop[ntime, ]==0) # Olhe pra última linha da matriz, o tempo final, e calcule quantas populações chegaram a ter o p=0. Esse é o número das populações que fixaram o alelo q.
  ### Criando o gráfico final ####
  par(mfrow=c(1,1)) #Para er certeza que será apenas um gráfico por janela gráfica
  for (po in 1:n) # Crio um ciclo pra sobrepor os gráficos, aí cada linha representa uma população, com a trajetória dos valores do p nessa população.
  {
    cores <- rainbow(n) # Selecione uma cor pra cada população.
    fim <- plot(seq(1:ntime), matriz.pop[ ,po], # Plote cada coluna/populaçao da matriz pelo tempo.
                family="mono", # Selecione essa fonte para o gráfico.
                xlab="Gerações (ntime)", ylab="Frequência de p", # Nomeie os eixos.
                xlim=c(1,ntime), ylim=c(0,1), # Coloque esses limites nos eixos.
                cex.axis=0.8, cex.lab=0.8, # Coloque esse tamanho nos eixos e nome dos eixos
                tcl=-0.3, las=1,  # Ajuste os "riscos" e posiçoes do eixo.
                bty="n", # Deixa apenas a escala nos eixos
                type="l", col=cores[po]) # Plote apenas a linha, com a sua devida cor.
    par(new=TRUE) # Adicione o outro gráfico, da próxima população, junto à esse.
  }
  points(1,freqi, pch=16, col="black") # Coloque um ponto no p inicial
  title(paste("In put: n=", n, ", npop=", npop, ", ntime=", ntime, ", freqi=", freqi, ", s=", s, " , sel.dom=", sel.dom, "\n \n Out put: p fixado=", fixp , " q fixado=", fixq, sep=" "), family="mono" ,cex.main=0.9) # Adicione o título ao gráfico, sendo que esse contém os valores de in put dos argumentos e, como out put, o número de populações que fixaram alelos e o próprio gráfico.

  ### Inserindo as linhas das trajetórias deterministas, ou seja, se não
```

```
houvesse o sample durante a reprodução ###
  if(s==0) # Se não houver seleção e não tivesse efeito de deriva:
  {
    lines(seq(1:ntime), rep(freqi, times=ntime), lty=2, lwd=1.7) # Plote a
linha com o valor de p sendo estático, sempre como a probabilidade de freqi.
  }
  if(sel.dom==FALSE) # Se houver seleção no recessivo, sem deriva:
  {
    linef <- rep(NA, times=ntime) # Faz um vetor, como se fosse apenas uma
população.
    linef[1] <- freqi # Coloca a freqi na primeira linha, que é o tempo 1.
    for(deter in 2:ntime) # Faça um ciclo pra preencher o resto no vetor.
    {
      pdet <- linef[deter-1] # Pegue o valor de p da posição anterior.
      qdet <- 1-pdet # Com isso você também tem o valor de q.
      calcdet <- (pdet^2)/(1-((qdet^2)*s)) + 0.5*((2*pdet*qdet)/(1-
((qdet^2)*s))) # Calcule o novo valor de p com a mesma formula derivada de
H-W usada anteriormente.
      linef[deter] <- calcdet # Coloque o valor de p calculado na posição do
vetor.
    }
    lines(seq(1:ntime), linef, lty=2, lwd=1.7) # Plote a linha com o valor
de p calculado, é o p de um cálculo "determinista"
  }
  if(sel.dom==TRUE) # Se houvesse seleção no dominante, sem deriva:
  {
    linet <- rep(NA, times=ntime) # Faz um vetor, como se fosse apenas uma
população.
    linet[1] <- freqi # Coloca a freqi na primeira linha, que é o tempo 1.
    for(deter in 2:ntime) # Faça um ciclo pra preencher o resto no vetor.
    {
      pdet <- linet[deter-1] # Pegue o valor de p da posição anterior.
      qdet <- 1-pdet # Com isso você também tem o valor de q.
      calcdet <- ((pdet^2)*(1-s)/(1-((pdet^2)*s)-(2*pdet*qdet*s))) +
0.5*((2*pdet*qdet*(1-s))/(1-((pdet^2)*s)-(2*pdet*qdet*s))) # Calcule o novo
valor de p com a mesma formula derivada de H-W usada anteriormente.
      linet[deter] <-calcdet # Coloque o valor de p calculado na posição do
vetor.
    }
    lines(seq(1:ntime), linet, lty=2, lwd=1.7) # Plote a linha com o valor
de p calculado, é o p de um cálculo "determinista"
  }
  par(new=FALSE) # Caso a pessoa faça a funçao rodar várias vezes, isso não
vai sobrepor todos os gráficos
  ### Pedindo pra retornar o gráfico na janela gráfica e FIM ###
  return(fim) # Retorne o gráfico
}
```

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

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2016:alunos:trabalho_final:louiseee.morais:start

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