

Laura Cristina Multini



Doutoranda em Entomologia em Saúde Pública pela Faculdade de Saúde Pública - USP. [FSP USP](#)

Trabalha atualmente com caracterização genética, fenotípica e ecologia da espécie de mosquito *Anopheles cruzii* que é responsável pela transmissão de patógenos que causam a malária em regiões de Mata Atlântica.

[exec](#)

Proposta

Estimativa de F-estatística para análise de estrutura populacional Os parâmetros de F-estatística (FST, FIS e FIT) estimados a partir de dados genéticos, introduzidos por Wright (1951) oferecem meios convenientes de resumir a estrutura genética de populações (Weir & Cockerham 1984). O cálculo de FST de Weir & Cockerham (1984), ainda permanece uma métrica útil, pois é imparcial ao tamanho da amostra e, provavelmente, continuará a ser usada nos próximos anos.

Função

A função que proponho tem como objetivo calcular as estimativas de FST com número de iterações a escolha do usuário para dados genéticos de populações obtidos a partir do tamanho do fragmento amplificado para vários *loci* polimórficos.

Os cálculos gerados pela função serão:

- Global para todos os *loci* em todas as populações.
- Para cada *locus* separadamente em todas as populações.
- Valores de *P* serão estimados para cada parâmetro testado.

Input

Data frame com tamanho de fragmentos amplificados para cada população, para todos os *loci* utilizados pelo usuário – ideal que não tenha limite de número de populações, nem de *loci* utilizados.

ExemploInput

Output

- semi-matrizes de diferenciação genética contendo estimativas de FST global (para todos os *loci* em todas as populações).
- semi-matrizes de diferenciação genética contendo estimativas de FST para cada *locus* em todas as populações.
- semi-matrizes equivalentes contendo os valores de *P* para todos os parâmetros testados.

Justificativa

Existem muitos programas e alguns pacotes no R que podem calcular as estimativas de FST de Weir & Cockerham rapidamente a partir de dados genéticos. Todos esses programas são extensamente utilizados e fazem um ótimo trabalho. Porém, realizar uma função com esses cálculos “na mão” pode ser particularmente útil e desafiador.

Plano “B”

Uma segunda proposta seria fazer uma função que plote gráficos com base nos resultados da proposta 1. Os gráficos gerados poderiam ser árvores de distância (UPGMA, Dendrograma ou Neighbor-Joining – com valores de bootstrap). O usuário poderia escolher qual gráfico plotar e suas configurações.

Input

- Resultados gerados pela função da proposta 1 (matriz de diferenciação genética, com valores iguais na diagonal de cima e de baixo).
- Data frame com tamanho de fragmentos amplificados para cada população, para todos os *loci* utilizados pelo usuário. Nesse caso, a função criada chamaria a função do pacote do R [diveRsity](#) que faz os cálculos de FST e plotaria os gráficos com base nesses resultados, o usuário poderia ter a opção da função mostrar os resultados com as semi-matrizes de diferenciação genética e os gráficos juntos ou separados.

Output

- gráficos de árvores de distância a escolha do usuário – UPGMA, Dendrograma, Neighbor-Joining.

Exemplo de gráfico:



Justificativa

A minha ideia principal era fazer uma função que calculasse os valores de FST da proposta 1 e plotaria os gráficos da proposta 2, porém analisando as dificuldades inerentes à construção da função 1, resolvi dividi-las em duas propostas diferentes. Os gráficos plotados sempre melhoram a visualização dos resultados de diferenciação genética entre populações.

Comentários Melina

Olá Laura,

eu não entendo de genética o suficiente para saber quão fácil ou difícil é criar uma função que calcula a FST “na mão”, ou seja, sem auxílio de funções já existentes ou sem “colar” de outras funções, mas acredito em você quando diz que será desafiador.

Minha sugestão vai ser que você consiga juntar as duas propostas em uma. Eu vi que você está receosa em fazer isso, mas parece que você domina o assunto suficientemente para ser capaz de fazer as duas e uma. Para os plots, claro, vale usar funções de plot existentes para facilitar. Isso porque, fazer a proposta B já baseada no resultado da proposta A (ou da função que já

existe) é simples demais e não demandará quase nada de código. Desta forma, seria legal você colocar um argumento na sua função para as possibilidades de gráficos de saída, que tal?

Qualquer coisa, estou à disposição.

[Melina Leite](#) 2017/06/07 14:57

Comentário Laura

Olá Melina, Muito obrigada pela sua resposta!

Acho que posso tentar colocar um argumento de saída de gráfico na minha função. Com certeza será um desafio, mas melhorará consideravelmente minha função.

Tentarei usar funções de plots para facilitar a saída da função - pelo menos essa parte.

Então está combinado. Começarei a trabalhar na função.

Obrigada, Laura

Arquivo para testar função `fst.fun`

[input](#)

Baixe o arquivo e salve no diretório de trabalho

Help da Função

`fst.fun`

package:nenhum

R Documentation

#Função para cálculo Global de FST para dados "multi-loci"

Description:

A função `fst.fun` calcula o FST Global para um número ilimitado de loci microssatélites para todas as populações possíveis.

Usage:

```
fst.fun(x)
```

Arguments:

`x` Obrigatoriamente um documento no formato `genepop` (".gen" ou ".txt" a função aceita as duas extensões) contendo todos os alelos, seus nomes e

populações que deve estar salva no diretório de trabalho.
Para exemplo (ver Details).

Details:

```
x
"Microsatellite"
Locus1
Locus2
Locus3
Locus4
Locus5
Locus6
Locus7
Locus8
POP
      POP1 , 196196 288288 090132 345417 120120 162162 174180 297459
      POP1 , 196196 286286 090132 414414 120120 162162 174180 294450
      POP1 , 196196 286286 090132 342414 120120 162162 174180 294459
      POP1 , 196196 288288 090132 417417 120120 162162 174180 294459
POP
      POP2 , 196196 286286 090132 372417 108108 162162 174180 294450
      POP2 , 196196 286286 090132 372411 120120 162162 174180 294459
      POP2 , 196196 286286 090130 372411 108108 162162 174180 294294
```

Value:

Matriz contendo o FST Global para todas as populações (linhas) em todos os loci (coluna).

Warning:

```
É necessário instalar o pacote diveRsity
install.packages("diveRsity", dep = TRUE)
```

Note:

```
#####
```

Author(s):

```
Laura Cristina Multini
lauramultini@usp.br
```

References:

Keenan, K., McGinnity, P., Cross, T.F., Crozier, W.W., & Prodöhl, P.A., (2013), diveRsity: An R package for the estimation of population genetics parameters and their associated errors, *Methods in Ecology and Evolution*, <doi:10.1111/2041-210X.12067>

Weir, B.S. & Cockerham, C.C., Estimating F-Statistics, for the Analysis of Population Structure, *Evolution*, vol. 38, No. 6, pp. 1358-1370 (1984).

See Also:

Pacote `diveRcity`
Functions: `fastDivPart`

Examples:

```
install.packages("diveRcity", dep = TRUE)
fst.fun("input.txt")
```

Código da função

```
#Função para cálculo Global de FST para dados "multi-loci"

fst.fun <- function(x)
{
  #Para rodar a função é necessário instalar o pacote diveRcity
  library(diveRcity)
  x <- readGenepop(x) #Transformando o input que deve ser obrigatoriamente
  ".gen ou .txt"

  if(class(x) == "list")
  {

#Calculo das frequencias do alelo "p" na pop "x"
p.calc <- list()
for(i in 1:x$nloci)
{
  p.calc[[i]] <- 2*(x$allele_freq[[i]][,]+x$allele_freq[[i]][,])/x$pop_sizes
}

#Calculo das frequencias do alelo "q" na pop "x"
q.calc <- list()

for(j in 1:x$nloci)
{
  q.calc[[j]] <- 1 - p.calc[[j]][,]
}

#Calculo da heterozigosidade esperada (HE)
h.exp <- list()

for(n in 1:x$nloci)
{
  h.exp[[n]] <- 1 - ((p.calc[[n]][,])^2+(q.calc[[n]][,])^2)
}

#Calculo da media de HE para todas as populações em todos os loci
media.hexp <- list()
```

```
for (k in 1:x$nloci)
{
  media.hexp[[k]] <- colMeans(h.exp[[k]][,], dims = 1)
}

data.hexp <- matrix(NA, nrow=x$npops, ncol=x$nloci)

data.hexp2=as.data.frame(data.hexp)

#Passando dados que estavam em lista para data.frame
for(l in 1:k)
{
  data.hexp2[,l] <- media.hexp[[l]]
}

ht <- matrix(NA, nrow = x$npops, ncol = x$nloci)

ht2 <- as.data.frame(ht)

#Calculo do HT - frequencia media do alelo "p" e "q" na pop
for(h in 1:x$nloci)
{
  ht2[[h]] <- 1 - ((mean(p.calc[[h]][,])^2)+(mean(q.calc[[h]][,])^2))
}
#Calculo do FST - HT-HS/HT
fst.teste <- (ht2-data.hexp2)/ht2

#Calculo da media
media.fst <- apply(fst.teste, 1, mean)

#Arrumando o output
names(media.fst)[1:x$npops] <- c(x$pop_names)

global <- as.matrix(media.fst)

colnames(global) <- c("FST$Global")

return(global)

}
}
```

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

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2017:alunos:trabalho_final:lauramultini:start

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

