Rodrigo dos Santos Francisco

Mestrado em Genética e Biologia Evolutiva, Instituto de Biosciencias, USP-SP Doutorado (em andamento) em Genética e Biologia Evolutiva, IB-USP

EXERCÍCIOS

.exec

TRABALHO FINAL

Plano A

Contexto biológico

Os genes HLA (antígenos leucocitários humanos) apresentam elevados níveis de polimorfismo e grande variação entre as diferentes populações humanas. Esses genes tem como função a apresentação de peptídeos antigênicos aos linfócitos T e interação com os receptores KIR das células natural killers, apresentando papel central na construção da resposta imune adaptativa. A impressionante diversidade dos alelos de HLA concentra-se nos éxons codificantes da região de ligação de peptídeo. Há uma série de evidências de que os genes HLA estejam evoluindo sob ação da seleção natural, entretanto, ainda há debate a respeito de como a seleção vem atuando. Uma das tentativas de entendimento dos mecanismos evolutivos atuantes nesse sistema foi realizada com a classificação dos alelos dos genes HLA-A e -B em Supertipos.

Supertipos são grupos de alelos que teriam capacidades redundantes de apresentação de antígenos. As moléculas HLA codificadas por alelos dentro de um mesmo supertipo partilhariam aminoácidos específicos nas posições de ancoragem dos peptídeos. De acordo com essa classificação, não seriam os alelos individuais, mas sim os supertipos, as unidades que responderiam à atuação da seleção balanceadora. Como predição gerada por essa hipótese, espera-se que as frequências dos supertipos seriam mantidas em níveis elevados e conservados entre diferentes etnias, a despeito da variação dos alelos individuais. Entretanto, é possível que o agrupamento de um grande número de alelos em poucos grupos (como são os supertipos) cause uma menor diferenciação entre as populações.

Proposta:

Minha proposta é a criação de uma função que permitiria a realização do teste da hipótese de que as populações seriam menos diferenciadas quando analisadas do ponto de vista dos supertipos. Nosso objetivo é verificar se o grau de diferenciação dos supertipos é menor do que seria observado sob a hipótese nula de que esses agrupamentos sejam aleatórios (não funcionais ou filogenéticos). Dessa forma a hipótese nula refletiria a ausência de função biológica como um determinante dos níveis de diferenciação inter-populacional

1) A idéia seria partir de um dataset como o exemplificado abaixo:

(tabela_01)

population	region	id	a_1	a_2	b_1	b_2
Adygei	EUR	JK3156	0201	0201	3503	5101
Adygei	EUR	JK3157	1101	3201	0702	5101
Adygei	EUR	JK3158	0301	3201	0702	0702
Spanish	EUR	JK3159	1101	2301	3501	4901
Spanish	EUR	JK3160	0101	1101	0801	1801
Mbuti	SAF	JK3161	0201	1101	1501	5101
Mbuti	SAF	JK3162	0201	3303	1402	5801

2) A primeira etapa envolveria a aplicação da classificação de supertipos, substituindo os alelos pelos seus respectivos supertipos:

A correspondência entre supertipos e alelos poderia ser dada em vetores (?????)

Ex.:

A01 ← (0101,0102,0103...1101)

3) Em seguida as frequências de supertipos, por população e/ou região seriam calculadas, permitindo o calculo da taxa de heterozigose.

4) Com as taxas de heterozigose eu pretendo calcular as medidas de diferenciação populacional Gst e D de Jost.

5) Eu quero que a saída seja uma matriz poulação/população, mostrando os valores dessas medidas entre cada par de população. E nessa matriz, os valores seriam substituídos por níveis de cores.

6) Esse seria o cálculo das medidas de diferenciação para o dados observado. O teste consistiria no embaralhamento dos elementos constituintes dos vetores que vinculam os supertipos aos alelos (eliminando a associação biológica). Com esses novos vetores, eu repetira todos os passos que citei nos itens anteriores, mas guardaria as medidas de diferenciação entre os pares de populações ao final. Esse embaralhamento seria repetido 1000 vezes, tendo ao final uma distribuição de valores de de Gst e D para cada par de população. Ao final, eu verificaria o número de pares de populações que apresentam níveis de diferenciação nas caudas de distribuição gerada pelos embaralhamentos.

PLANO B

O Plano B consistiria nas etapas 1 a 6, descritas acima. Com elas eu conseguiria calcular as medidas de diferenciação populacional e plotaria o resultado na matriz como comentei, mas não faria a simulação.

Comentários das Propostas (Leo)

A proposta parece interessante, mas um pouco trabalhosa. Sugiro trabalhar na proposta B (passos 1 a 5), deixando a simulação de fora, que pode ser implementada no futuro conforme tua necessidade e disponibilidade.

Página de Ajuda

```
d.pop                        package:NA                    R Documentation


Diferenciação Populacional


Description:

Função que calcula os valores de Gst e D de Jost, duas medidas de
diferenciação populacional para dados genéticos. A função retorna uma matriz
numérica e uma figura de grade de  cores para cada uma dessas estatísticas,
mostrando os valores de Gst e D entre cada par de populações analisadas.


Usage:

d.pop(x)


Arguments:

x: objeto da classe data.frame

population: Lógico, se verdadeiro (TRUE)(default), realiza o cálculo de Gst
e D de Jost entre cada par de populações individuais.
ex: d.pop(x) = d.pop(x, population=T)

Se falso (FALSE), realiza o cálculo de Gst e D de Jost entre os grupos de
populações especificados na coluna "region" do data.frame (agrupando as
populações por regiões).
ex: d.pop(x, population=F)


Details:


x: objeto da classe data.frame. DEVE CONTER 5 COLUNAS CORRESPONDENTES ÀS
SEGUINTES INFORMAÇÕES:
coluna 1: nomes das populações de onde os indivíduos são provenientes
coluna 2: nome dos agrupamentos de populações (regiões),
coluna 3: o id dos indivíduos,
coluna 4: alelo 1 e
coluna 5: alelo 2.


IMPORTANTE:
1) O data.frame deve conter um cabeçário. A leitura do arquivo para geração
do data.frame deve levar o cabeçário em consideração:
```

```
ex.: read.table(x,header=TRUE);

2) A presença de NAs é permitida, contudo, todas as linhas contendo NAs
serão removidas do data.frame;

3) Com a ausência de uma das colunas haverá o retorno de uma mensagem de
erro:
"ERRO: O objeto de entrada para a função d.pop deve ser obrigatoriamente um
data.frame com 5 colunas contendo informações na seguinte ordem: população,
região (conjunto de populações), id dos indivíduos, alelo 1 e alelo 2. Por
favor, verifique o número de colunas do data.frame"


Value:

A função d.pop  retornará a seguinte mensagem:

"As Matrizes contendo os valores de Gst e D de Jost entre cada par de
populações foram geradas com sucesso e salvas nos arquivos Gst.txt e D.txt,
respectivamente. Além disso, foi gerado um gráfico para cada uma das
matrizes. Todos esses arquivos estão disponíveis no diretório: "nome do
diretório da area de trabalho""

Gst.txt: contém a matriz com os valores de Gst entre cada par de populações
(uma matriz par a par);

D.txt: contém a matriz com os valores de D de Jost entre cada par de
populações (uma matriz par a par);

Para cada uma das matrizes é gerada uma figura na forma de grade de cores;


WARNING:

A função d.pop irá substituir qualquer arquivo com o nome Gst.txt e D.txt
presente no diretório da área de trabalho, assim como as figuras geradas a
partir das matrizes;



Author(s):

Rodrigo dos Santos Francisco <biorodrigo2001@yahoo.com.br>

Collaborator(s):

Barbara D. Bitarello

Maria Helena T. Maia
```

```
Débora Y. C. Brandt

References:

Jost, Lou. 2008. “G ST and its relatives do not measure differentiation.”
Molecular Ecology 17(18): 4015-4026.

See Also:

Examples:

#Utilizar o arquivo de exemplo no fim da página.
Exemplo 1: d.pop(x)
Exemplo 2: d.pop(x, population=F)
```

Código da Função

```
d.pop <- function(x,population=T)
    {
#Retirada de NAs e geraçao das listas que serão utilizadas na função

if(dim(x)[2]!=5)
        {
        cat("\n\n\n ERRO: O objeto de entrada para a função d.pop deve ser
obrigatoriamente um data.frame com 5 colunas contendo informações na
seguinte ordem: população, região (conjunto de populações), id dos
indivíduos, alelo 1 e alelo 2. Por favor, verifique o número de colunas do
data.frame.\n\n\n")
        }
    else
        {
    tabela <-(na.omit(x))
    a <- list()
    b <- list()
    d <- list()
    f<-list()
    fabs<-list()
    frel<-list()
    H <- NA

######## Trabalhando-se com as populações individuais-----------------------
--------
    if(population==T)
        {
        population <- data.frame(tabela[,1],tabela[,4],tabela[,5])
        v.population <- unique(c(as.character(population[,1])))
        population[,1]<- as.factor(c(population[,1]))

# Geração do contador-----------------------
```

```
        n <- nlevels(population[,1])
        loop<-(n*(n-1))/2
        count<-rep(2:n,c(seq(from=1,to=n-1)))
        count2<-1
        for(i in 3:n-1)
            {
            count2=c(count2,seq(1:i))
            }
        counter <- cbind(count, count2)
# Geração de Matrizes vazias para guardar os resultados dos ciclos a seguir-
---------------------------------------
        Hs <- matrix(,n,n)
        rownames(Hs)<-v.population
        colnames(Hs)<-v.population
        HT <- matrix(,n,n)
        rownames(HT)<-v.population
        colnames(HT)<-v.population
####### Ciclos para o cálculo das Taxas de Heterozigose Intra-Populacionais-
-----------------------------------

        for(i in 1:n) # Separação das populações;
            {
            a[[i]] <- (subset(population,tabela...1.== i, select =
c(tabela...4.,tabela...5.)))
            }
        for(i in 1:n) # Cálculo das frequências absolutas dos alelos;
            {
            b[[i]] <-
as.data.frame(summary(as.factor(c(as.character(a[[i]]$tabela...4.),as.charac
ter(a[[i]]$tabela...5.)))))
            }
        for(i in 1:n) # Cálculo das frequências relativas;
            {
            d[[i]] <- b[[i]][,1]/sum(b[[i]][,1])
            }
        for(i in 1:n) # Cálculo das Taxas de Heterozigose Intra-
Populacionais;
            {
            H[[i]] <- 1-(sum((d[[i]])^2))
            }
        for(i in 1:n) # Cálculo das Taxas de Heterozigose Médias entre cada
par de população;
            {
            for(j in 1:nlevels(population[,1]))
                {
                Hs[i,j] <- mean(c(H[i],H[j]))
                }
            Hs[upper.tri(Hs, diag=T)] <- NA
            }
####### Ciclos para o cálculo da Taxa de Heterozigose TOTAL-----------------
```

```
-------------------
        for(i in 1:loop) # Criação das amalgamas de populações (par a par)
            {
            f[[i]] <- rbind(a[[counter[i,1]]],a[[counter[i,2]]])
            }
        for(i in 1:loop) # Cálculo das frequências absolutas dos alelos;
            {
            fabs[[i]] <-
as.data.frame(summary(as.factor(c(as.character(f[[i]]$tabela...4.),as.charac
ter(f[[i]]$tabela...5.)))))
            }
        for(i in 1:loop) # Cálculo das frequências relativas;
            {
            frel[[i]] <- fabs[[i]][,1]/sum(fabs[[i]][,1])
            }
        for(i in 1:loop) # Cálculo da Taxa de Heterozigose Total para cada
par de população;
            {
            HT[counter[i,1],counter[i,2]] <- 1-(sum((frel[[i]])^2))
            }
####### Cálculo das estatísticas Gst e D de Jost----------------------------
--------
        Gst <- (HT-Hs)/HT
        D <- (HT-Hs)/((1-Hs)*(n/(n-1)))
####### Geração do Gráfico de Gst---------------------------
        ColorRamp <- colorRampPalette(c("white", "steelblue1", "blue3"))
        outfileGraphic <- paste("pair_Gst_Matrix","population",Sys.Date() ,
".png", sep="_")
        png(outfileGraphic, width=1300, height=1300, res=144)
        smallplot <- c(0.874, 0.9, 0.18, 0.83)
        bigplot <- c(0.13, 0.85, 0.14, 0.87)
        old.par <- par(no.readonly = TRUE)
# Legenda --------------------------------
        par(plt = smallplot)
        Min <- min(Gst, na.rm=TRUE)
        Max <- max(Gst, na.rm=TRUE)
        binwidth <- (Max - Min) / 64
        y <- seq(Min + binwidth/2, Max - binwidth/2, by = binwidth)
        z <- matrix(y, nrow = 1, ncol = length(y))
        image(1, y, z, col = ColorRamp(64),xlab="", ylab="", axes=FALSE)
        axis(side=4, las = 2, cex.axis=0.8)
        box()
        mtext(text=expression(bold(Fst)), side=4, line=2.5, cex=1.1)
# Gráfico propriamente dito ---------------------------

        e <- ncol(D)
        f <- nrow(D)
        x <- c(1:e)
        y <- c(1:f)
        par(new = TRUE, plt = bigplot)
        image(x,y,Gst, col=ColorRamp(64),
```

```
main=expression(bold(Matrix~of~pairwise~Gst)), xlab="",ylab="", axes=FALSE)
        box()
        axis(1, at = c(1:e), labels=c(v.population), cex.axis=0.75,las=2)
        axis(2, at = c(1:f), labels=c(v.population), cex.axis=0.75,las=2)
        par(old.par)
        dev.off()
####### Geração do Gráfico de D de Jost---------------------------
        ColorRamp <- colorRampPalette(c("white", "steelblue1", "blue3"))
        outfileGraphic <- paste("pair_D_Matrix","population",Sys.Date() ,
".png", sep="_")
        png(outfileGraphic, width=1300, height=1300, res=144)
        smallplot <- c(0.874, 0.9, 0.18, 0.83)
        bigplot <- c(0.13, 0.85, 0.14, 0.87)
        old.par <- par(no.readonly = TRUE)
# Legenda --------------------------------
        par(plt = smallplot)
        Min <- min(D, na.rm=TRUE)
        Max <- max(D, na.rm=TRUE)
        binwidth <- (Max - Min) / 64
        y <- seq(Min + binwidth/2, Max - binwidth/2, by = binwidth)
        z <- matrix(y, nrow = 1, ncol = length(y))
        image(1, y, z, col = ColorRamp(64),xlab="", ylab="", axes=FALSE)
        axis(side=4, las = 2, cex.axis=0.8)
        box()
        mtext(text=expression(bold(D)), side=4, line=2.5, cex=1.1)
# Gráfico propriamente dito ---------------------------

        e <- ncol(D)
        f <- nrow(D)
        x <- c(1:e)
        y <- c(1:f)
        par(new = TRUE, plt = bigplot)
        image(x,y,D, col=ColorRamp(64),
main=expression(bold(Matrix~of~pairwise~D)), xlab="",ylab="", axes=FALSE)
        box()
        axis(1, at = c(1:e), labels=c(v.population), cex.axis=0.75,las=2)
        axis(2, at = c(1:f), labels=c(v.population), cex.axis=0.75,las=2)
        par(old.par)
        dev.off()

###### Saída--------------------

    write.table(Gst, "Gst.txt", sep="\t", row.names=T)
    write.table(D,"D.txt", sep="\t", row.names=T)
    cat("\n\n\n\t As Matrizes contendo os valores de Gst e D de Jost entre
cada par de populações foram geradas com sucesso e salvas nos arquivos
Gst.txt e D.txt, respectivamente. Além disso, foi gerado um gráfico para
cada uma das matrizes. Todos esses arquivos estão disponíveis no
diretório:\n\n\n")
    return(getwd())
```

```
        }

######## Trabalhando-se com conjunto de populações (Regiões)----------------
---------------

    else
        {
        population <- data.frame(tabela[,2],tabela[,4],tabela[,5])
        v.population <- unique(c(as.character(population[,1])))
        population[,1]<- as.factor(c(population[,1]))
# Geração do contador-----------------------
        n <- nlevels(population[,1])
        loop<-(n*(n-1))/2
        count<-rep(2:n,c(seq(from=1,to=n-1)))
        count2<-1
        for(i in 3:n-1)
            {
            count2=c(count2,seq(1:i))
            }
        counter <- cbind(count, count2)
# Geração de Matrizes vazias para guardar os resultados dos ciclos a seguir-
---------------------------------------
        Hs <- matrix(,n,n)
        rownames(Hs)<-v.population
        colnames(Hs)<-v.population
        HT <- matrix(,n,n)
        rownames(HT)<-v.population
        colnames(HT)<-v.population
####### Ciclos para o cálculo das Taxas de Heterozigose Intra-Populacionais-
-----------------------------------

        for(i in 1:n) # Separação das populações;
            {
            a[[i]] <- (subset(population,tabela...2.== i, select =
c(tabela...4.,tabela...5.)))
            }
        for(i in 1:n) # Cálculo das frequências absolutas dos alelos;
            {
            b[[i]] <-
as.data.frame(summary(as.factor(c(as.character(a[[i]]$tabela...4.),as.charac
ter(a[[i]]$tabela...5.)))))
            }
        for(i in 1:n) # Cálculo das frequências relativas;
            {
            d[[i]] <- b[[i]][,1]/sum(b[[i]][,1])
            }
        for(i in 1:n) # Cálculo das Taxas de Heterozigose Intra-
Populacionais;
            {
            H[[i]] <- 1-(sum((d[[i]])^2))
            }
```

```
        for(i in 1:n) # Cálculo das Taxas de Heterozigose Médias entre cada
par de população;
            {
            for(j in 1:nlevels(population[,1]))
                {
                Hs[i,j] <- mean(c(H[i],H[j]))
                }
            Hs[upper.tri(Hs, diag=T)] <- NA
            }
####### Ciclos para o cálculo da Taxa de Heterozigose TOTAL----------------
-------------------
        for(i in 1:loop) # Criação das amalgamas de populações (par a par)
            {
            f[[i]] <- rbind(a[[counter[i,1]]],a[[counter[i,2]]])
            }
        for(i in 1:loop) # Cálculo das frequências absolutas dos alelos;
            {
            fabs[[i]] <-
as.data.frame(summary(as.factor(c(as.character(f[[i]]$tabela...4.),as.charac
ter(f[[i]]$tabela...5.)))))
            }
        for(i in 1:loop) # Cálculo das frequências relativas;
            {
            frel[[i]] <- fabs[[i]][,1]/sum(fabs[[i]][,1])
            }
        for(i in 1:loop) # Cálculo da Taxa de Heterozigose Total para cada
par de população;
            {
            HT[counter[i,1],counter[i,2]] <- 1-(sum((frel[[i]])^2))
            }
####### Cálculo das estatísticas Gst e D de Jost---------------------------
--------
        Gst <- (HT-Hs)/HT
        D <- (HT-Hs)/((1-Hs)*(n/(n-1)))
####### Geração do Gráfico de Gst---------------------------
        ColorRamp <- colorRampPalette(c("white", "steelblue1", "blue3"))
        outfileGraphic <- paste("pair_Gst_Matrix","region",Sys.Date() ,
".png", sep="_")
        png(outfileGraphic, width=1300, height=1300, res=144)
        smallplot <- c(0.874, 0.9, 0.18, 0.83)
        bigplot <- c(0.13, 0.85, 0.14, 0.87)
        old.par <- par(no.readonly = TRUE)
# Legenda --------------------------------
        par(plt = smallplot)
        Min <- min(Gst, na.rm=TRUE)
        Max <- max(Gst, na.rm=TRUE)
        binwidth <- (Max - Min) / 64
        y <- seq(Min + binwidth/2, Max - binwidth/2, by = binwidth)
        z <- matrix(y, nrow = 1, ncol = length(y))
        image(1, y, z, col = ColorRamp(64),xlab="", ylab="", axes=FALSE)
```

```
        axis(side=4, las = 2, cex.axis=0.8)
        box()
        mtext(text=expression(bold(Fst)), side=4, line=2.5, cex=1.1)
# Gráfico propriamente dito ---------------------------

        e <- ncol(D)
        f <- nrow(D)
        x <- c(1:e)
        y <- c(1:f)
        par(new = TRUE, plt = bigplot)
        image(x,y,Gst, col=ColorRamp(64),
main=expression(bold(Matrix~of~pairwise~Gst)), xlab="",ylab="", axes=FALSE)
        box()
        axis(1, at = c(1:e), labels=c(v.population), cex.axis=0.75,las=2)
        axis(2, at = c(1:f), labels=c(v.population), cex.axis=0.75,las=2)
        par(old.par)
        dev.off()
####### Geração do Gráfico de D de Jost---------------------------
        ColorRamp <- colorRampPalette(c("white", "steelblue1", "blue3"))
        outfileGraphic <- paste("pair_D_Matrix","region",Sys.Date() ,
".png", sep="_")
        png(outfileGraphic, width=1300, height=1300, res=144)
        smallplot <- c(0.874, 0.9, 0.18, 0.83)
        bigplot <- c(0.13, 0.85, 0.14, 0.87)
        old.par <- par(no.readonly = TRUE)
# Legenda --------------------------------
        par(plt = smallplot)
        Min <- min(D, na.rm=TRUE)
        Max <- max(D, na.rm=TRUE)
        binwidth <- (Max - Min) / 64
        y <- seq(Min + binwidth/2, Max - binwidth/2, by = binwidth)
        z <- matrix(y, nrow = 1, ncol = length(y))
        image(1, y, z, col = ColorRamp(64),xlab="", ylab="", axes=FALSE)
        axis(side=4, las = 2, cex.axis=0.8)
        box()
        mtext(text=expression(bold(D)), side=4, line=2.5, cex=1.1)
# Gráfico propriamente dito ---------------------------

        e <- ncol(D)
        f <- nrow(D)
        x <- c(1:e)
        y <- c(1:f)
        par(new = TRUE, plt = bigplot)
        image(x,y,D, col=ColorRamp(64),
main=expression(bold(Matrix~of~pairwise~D)), xlab="",ylab="", axes=FALSE)
        box()
        axis(1, at = c(1:e), labels=c(v.population), cex.axis=0.75,las=2)
        axis(2, at = c(1:f), labels=c(v.population), cex.axis=0.75,las=2)
        par(old.par)
        dev.off()
###### Saída--------------------
```

```
        write.table(Gst, "Gst.txt", sep="\t", row.names=T)
        write.table(D,"D.txt", sep="\t", row.names=T)
        cat("\n\n\n\t As Matrizes contendo os valores de Gst e D de Jost
entre cada par de populações foram geradas com sucesso e salvas nos arquivos
Gst.txt e D.txt, respectivamente. Além disso, foi gerado um gráfico para
cada uma das matrizes. Todos esses arquivos estão disponíveis no
diretório:\n\n\n")
        return(getwd())
        }
    }
    }
```

Arquivo de Exemplo: populacoes.txt

Código da função: funcao_final.r