```
#funcao para calculo de desequilibrio de ligacao entre duas SNPs

#argumentos da funcao: X = objeto dos dados no formato "map" ; Y = objeto
dos dados no formato "ped"; snp1 e snp2 = nome das SNPS a serem analisadas.

DL <- function(x,y, snp1, snp2)

{
  #criar arquivo com os nomes das snps do arquivo map
  names.map <- x[,2, drop=FALSE]
  #nomear a coluna com os nomes das SNPs
  colnames(names.map) <- "SNP"
  #criar arquivo somente com os dados dos genotipos do arquivo ped
  genotipo.ped  <-y[, 7:length(y)]
  #nomear as colunas com os genotipos com o nome das respectivas SNPs
  colnames(genotipo.ped) <- paste(rep(names.map[,1], each=2), c("a", "b"),
sep="")
  #cria lista com nome das snps selecionadas pelo usuário
  inf<-c(snp1,snp2)
  #cria lista com o nome das snps selecionadas pelo usuário
  sel<-paste(rep(inf,each=2),c("a","b"), sep="")
  #cria arquivo para armazenar dados gerados no loop
  snpcl<-c()
  #loop para selecionar as colunas que serão lidas no próximo loop
  for(j in seq(1,length(sel),by=2)){
    #loop para selecionar as linhas que serão lidas no loop
    for(i in 1:dim(genotipo.ped)[1]) {
      #função para identificar nas os valores faltantes no arquivo (0) e
substiruir por NA
      snpcl =
c(snpcl,ifelse(genotipo.ped[i,sel][j]==0|genotipo.ped[i,sel][j+1]==0,NA,
                              #função para identificar alelos iguais ao alelo
do segundo nivel nas duas colunas, quando essa condição for verdadeira
substituir as duas linhas pelo velor 0.
ifelse(genotipo.ped[i,sel][j]==levels(genotipo.ped[,j])[2] &
genotipo.ped[i,sel][j+1]==levels(genotipo.ped[,j])[2],0,
                                      #função para identificar alelos que são
diferentes ao alelo do segundo nível nas duas colunas, quando essa condição
for verdadeira substituir por 2 e quando for falsa substituir por 1.
ifelse(genotipo.ped[i,sel][j]!=levels(genotipo.ped[,j])[2] &
genotipo.ped[i,sel][j+1]!=levels(genotipo.ped[,j])[2],2,1))
      ))
    }
  }
  #transformar os dados de saída do loop em uma matriz
  snp.m <- matrix(snpcl, ncol = length(sel)/2,)
  #calcular a frequencia de todos os gametas possiveis na população.
  N00 <-length(which(snp.m[,1]==0 & snp.m[,2]==0))
  N01 <-length(which(snp.m[,1]==0 & snp.m[,2]==1))
  N02 <-length(which(snp.m[,1]==0 & snp.m[,2]==2))
  N10 <-length(which(snp.m[,1]==1 & snp.m[,2]==0))
```

```
  N11 <-length(which(snp.m[,1]==1 & snp.m[,2]==1))
  N12 <-length(which(snp.m[,1]==1 & snp.m[,2]==2))
  N20 <-length(which(snp.m[,1]==2 & snp.m[,2]==0))
  N21 <-length(which(snp.m[,1]==2 & snp.m[,2]==1))
  N22 <-length(which(snp.m[,1]==2 & snp.m[,2]==2))
  #omitir os valores de NA dos calculos finais
  NNa <- length(na.omit(snp.m))
  #valor do total de genotipos
  total.pop <-length(snp.m)
  #total de individuos genotipados
  N.final <- (total.pop - NNa)/2
  #calculo da frequencia dos gametas na população
  X11 = (N00+N01/2+N10/2+N11/4)/(N.final)
  X12  = (N02+N01/2+N12/2+N11/4)/(N.final)
  X21  = (N20+N21/2+N10/2+N11/4)/(N.final)
  X22  = (N22+N21/2+N12/2+N11/4)/(N.final)
  #calculo da frequência dos alelos na população
  p1 = X11+X12
  p2 = X21+X22
  q1 = X11+X21
  q2 = X12+X22
  #calculo do desequlibrio de ligação
  D = (X11*X22)-(X12*X21)
  #calculo do Dmax
  if(D >=0){
    if (X12<X21){
      Dmax = X12
    }
    else
    {
      Dmax = X21
    }
  }
  if (D < 0)
  {
    if (-X11>-X22)
    {
      Dmax=-X11
    }
    else
    {
      Dmax=-X22
    }
  }
  #calculo do valor de D'
  Dl= D/Dmax
  #calculo do valor de correlação de desequilibrio de ligação
  r2 = D^2/(p1*p2*q1*q2)
 #calculo da frequencia esperada dos gametas na população
  esp <- data.frame(X11.o = p1*q1, X12.o = p1*q2, X21.o = p2*q1, X22.o =
```

```
p2*q2)
 #calculo da frequencia observada dos gametas na população
  obs <- data.frame(X11.e = p1*q1 + D, X12.e = p1*q2 - D, X21.e = p2*q1 - D,
X22.e = p2*q2 + D)

 #calculo do qui-quadrado
  X2 = sum(((obs[1:4]-esp[1:4])/esp[1:4])^2)
  resultado <- data.frame(D, Dl, r2, X2)
  names(resultado)<-c("D", "D´", "r²", "qui-quadrado")
  return(resultado)
}
```

```
modelo                package:unknown                R Documentation

function to calculate linkage disequilibrium (DL)

Description:

     The function calculate linkage disequelibrium (D) between two SNPs, the
function still returns the values of the

adjusted linkage disequilibrium (D´), the correlation between two SNPs (r²)
and the value of chi-square (X²).
     The imput data is compatible with Plink format, map and ped files.

Usage:

     DL (x, y, snp1, snp2)

Arguments:

 x - object of a file .map
 y - object of a file .ped
 snp1 - name of one SNP to be analysed. Must be included in file.map
 snp2 - name of the second SNP to be analysed. Must be included in file.map

Details:

The files must be in .map and .ped format and is no possible run the
function with any of arguments=NULL.
The files can contain differents chromossomes.


Value:

   For DL the results are a data.frame


Author(s):
```

```
     Carolina Mancini

References:

 Plink documentation. Shaun Purcell. May 2010

See Also:

     genetics package for more functions for genetic population

Examples:

teste.map <- data.frame(V1=rep(1, 5),V2=paste("rs", 12345:12349, sep=""),
V3=rep(0,5),V4=395679:395683)

teste.ped <-data.frame(V1=rep(1:75, each=2), V2=1101:1250, V3=rep(0,150),
V4=rep(0,150), V5=rep(1,2, each=2,75), V6=rep(-

9, 150),
                       V7=sample(rep(c("A","G",0),50)),
V8=sample(rep(c("G","A",0),50)),
                       V9=sample(rep(c("T","C",0),50)),
V10=sample(rep(c("C","T",0),50)),
V11=sample(rep(c("A","C",0),50)),V12=sample(rep(c("c","A",0),50)),
V13=sample(rep(c("G","T",0),50)),V14=sample(rep(c("T","G",0),50)),
V15=sample(rep(c("A","G",0),50)),V16=sample(rep(c("G","A",0),50)))

   DL(teste.map, teste.ped, "name snp1", "name snp2")
```

Proposta 1: Calcular o desequilíbrio de ligação entre dois alelos em uma determinada população. A função irá calcular os valores de D’, r^2 , a frequência estimada dos haplótipos, a frequência esperada em caso de equilíbrio de Hardy-Weinberg e o resultado do teste qui-quadrado. Os dados de entrada serão do tipo data.frame e poderão ser lidos a partir de um arquivo txt. O cálculo de LD será feito de forma pareada, o usuário poderá selecionar os alelos de interesse ou fazer a análise para todos os alelos do arquivo. Também será possível selecionar os alelos selecionando a região genômica, nesse caso o data.frame deverá conter informações sobre a posição física (bp) dos alelos. Os dados de saída estão na forma matriz contendo os valores de D’, r^2 e χ^2 para cada par de alelos testados. Os resultados da frequência estimada dos haplótipos e da frequência esperada serão visualizados apenas quando o usuário escolher apenas dois alelos o cálculo do LD. Nesse caso o resultado será um data.frame. w05_lecture15.pdf

Proposta 2: Determinação de modelos de regressão logística para fenótipos complexos.

A função irá determinar quais as variáveis estão relacionadas com o fenótipo. Todas as possíveis variáveis preditoras serão testadas, e serão selecionadas as que apresentarem modelo com p-value ≥0.05. As variáveis selecionadas serão submetidas a um teste para identificar colinearidade entre elas. Por fim será construído um modelo de regressão logística com base nas variáveis selecionadas. Os dados de entrada serão em forma de data.frame a partir de um arquivo txt e os dados de saída serão um data.frame contendo o modelo de regressão que melhor explica o fenótipo em questão e um gráfico com os dados e o modelo de regressão.

Carolina, a proposta 1 é bastante interessante. Você irá fazer o cálculo dos índices propostos na unha? Não esqueça de definir a entrada dos dados no help da sua função. A proposta 2 está um pouco confusa. Como são os dados de entrada? Qual modelo você irá usar na primeira etapa? — Juliana Vendrami 2015/03/24 15:05

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

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2015:alunos:trabalho_final:carol.mancini:trabalho_final

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