

Mendel

```

mendel=function(loci, modelo, N=F)#Cria a função "mendel" com os argumentos
loci, modelo e N
{
  install.packages("gplots")
  library(gplots)#Carregar pacote gplots para balloon plot
  #VERIFICANDO OS PARAMETROS
  if (modelo!= "t" && modelo != "s") #Se o argumento "modelo" for diferente
de "s" ou "t"
  {
    stop (" o argumento modelo deve ser 't' ou 's'") # Interrompe a funcao e
exibe a mensagem ao usuario
  }
  if (N!=FALSE)#Se N for diferente de FALSE
  {
    if (N != round (N) | N <= 0) #Se N nao for um numero inteiro e menor que
0
    {
      stop ("N deve ser um numero inteiro maior que 0.") #Interrompe a
funcao e exibe a mensagem ao usuario
    }
  }
  if (loci !=1 && loci!= 2) #Se o argumento loci for diferente de 1 e
diferente de 2
  {
    stop (" O argumento loci deve ser 1 ou 2") #Interrompe a funcao e exibe
mensagem ao usuario.
  }
  #CRIANDO OBJETOS COMUNS PARA OS DOIS MODELOS
  n.gametas=2^loci#Para calcular a quantidade de gametas possiveis
  ger1a.1=c(1,1)#Genótipo do progenitor 1 da primeira geração de 1 locus
(números se referindo à "AA")
  ger1b.1=c(3,3)#Genótipo do progenitor 2 da primeira geração de 1 locus
(números se referindo à "aa")
  ger1a.2=c(10,10)#Genótipo do progenitor 1 da primeira geração do segundo
locus (números se referindo à "BB")
  ger1b.2=c(20,20)#Genótipo do progenitor 1 da primeira geração do segundo
locus (números se referindo à "bb")
  AB=11#Cria objeto com o valor do gameta AB
  Ab=21#Cria objeto com o valor do gameta Ab
  aB=13#Cria objeto com o valor do gameta aB
  ab=23#Cria objeto com o valor do gameta ab
  ger2a.1= c(ger1a.1[1], ger1b.1[1])#genótipo do progenitor 1 oriundo da F1
(Aa), ou seja, um alelo de cada pai da geração parental
  ger2b.1= c(ger1a.1[2], ger1b.1[2])##genótipo do progenitor 2 oriundo da F1
(Aa)
  ger2a.2=c(AB, Ab, aB, ab)##Cria vetor com os gametas possiveis para o
progenitor 1 da f2(2 loci)
  ger2b.2=c(AB, Ab, aB, ab)#Cria vetor com os gametas possiveis para o

```

progenitor 2 da f2(2 loci)

```
AA=0# objeto com valor zero para servir de contador de frequencia de 'AA'
Aa=0# objeto com valor zero para servir de contador de frequencia de 'Aa'
aa=0# objeto com valor zero para servir de contador de frequencia de 'aa'
AABB=0#Cria objeto para contar a frequencia de AABB
#MODELO TEORICO
if (modelo=="t")#Se o argumento "modelo" for igual a "t", roda o modelo
teorico
{
  if (loci==1)#Se loci for igual 1, fazer cálculo de monohibridismo
  {
    punnet=matrix (NA, n.gametas,n.gametas, dimnames = list(c("a", "a"),
c("A", "A")))#Montar matriz com NAs para preencher com os valores, com
número de linhas e colunas de acordo com número de gametas
    for (j in 1:nrow(punnet))#fluxo "for" com contatdor j para linhas da
matriz "punnet"
    {
      for (i in 1:ncol(punnet))#fluxo "for" com contador i para colunas da
matriz "punnet"
      {
        punnet[j,i]= ger1a.1[i]+ ger1b.1[j]#Somar valores correspondentes ao
"A" do progenitor 1 com "a" do progenitor 2 e guardar em cada posição da
matriz
      }
    }
  }
  # #CONVERTER VALORES NUMERICOS EM LETRAS PARA O USUARIO
  for (j in 1:nrow(punnet)) #Fluxo 'for' com contador j para cada linha de
punnet
  {
    for (i in 1:ncol(punnet) )
    {
      if (punnet[j,i]==4)#Se a soma de alguma posição for igual a 4
      {
        punnet[j,i]="Aa"#substitui por "Aa"
      }
      if (punnet[j,i]==6)#Se a soma de alguma posição for igual a 6
      {
        punnet[j,i]="aa"#substitui por "aa"
      }
      if (punnet[j,i]==2)#Se a soma de alguma posição for igual a 2
      {
        punnet[j,i]="AA"#substitui por "AA"
      }
    }
  }
}
```

```

    }
  }
}
#SEGUNDA GERAÇÃO
punnet2=matrix (NA, n.gametas,n.gametas, dimnames = list(c("A", "a"),
c("A", "a")))#Montar matriz com NAs para preencher com oos valores, com
número de linhas e colunas de acordo com número de gametas
for (j in 1:nrow(punnet2))#Fluxo "for" com contador j para linhas da
matriz "punnet2"
{
  for (i in 1:ncol(punnet2))#fluxo "for" com contador i para colunas da
matriz "punnet2"
  {
    punnet2[j,i]=ger2a.1[i]+ger2b.1[j]#soma um alelo do progenitor 1 a
um alelo do progeitor 2 e guarda em uma posição da matriz punnet2
  }
}
#CONVERTER VALORES NÚMERICOS EM LETRAS PARA O USUARIO
for (j in 1:nrow(punnet2)) #Fluxo 'for' com contador j para cada linha
de punnet2
{
  for (i in 1:ncol(punnet2) )
  {
    if (punnet2[j,i]==4)#Se a soma de alguma posição for igual a 4
    {
      Aa = Aa + 1#soma 1 no objeto Aa
      punnet2[j,i]="Aa"#substitui por "Aa"
    }
    if (punnet2[j,i]==6)#Se a soma de alguma posição for igual a 6
    {
      aa= aa+1#soma 1 no objeto aa
      punnet2[j,i]="aa"#substitui por "aa"
    }
    if (punnet2[j,i]==2)#Se a soma de alguma posição for igual a 2
    {
      AA=AA+1#soma 1 no objeto AA
      punnet2[j,i]="AA" #substitui por "AA"
    }
  }
}
teorico.a=c(AA, Aa , aa)#Cria vetor com os valores contados nos objetos
AA, Aa e aa durante o fluxo acima
dominante=AA+Aa# Cria objeto para guardar os cotadores correspondentes
aos fenotipos dominantes
recessivo=aa#Cria objeto para guardar os cotadores correspondentes ao
fenotipo recessivo
fenotipos=c(dominante, recessivo)#Cria objeto que guarda ambos os
fenótipos
fenotipos.m=matrix(fenotipos, 1,2, dimnames = list(c("Freq"),
c("Dominante(A_)", "Recessivo(aa)")))#objeto que cria uma matriz com os
dados

```

```
fenotipos.t=as.table(fenotipos.m)# objeto que guarda uma tabela com os
fenotipos
par(mfrow= c(1,2), tcl=0.8, family="serif")# Divide o dispositivo de
tela em duas colunas.
barplot(teorico.a, names.arg=c("AA", "Aa", "aa"), col="coral3",
cex.lab=1.3, pch=19, main = "Proporção genotípica")#Barplot com as
frequencias genotipicas da f2
balloonplot(fenotipos.t, label =T, show.margins = F, dotcolor =
"coral3", main=" Proporção fenotípica")#balloonplot com as frequencias
fenotipicas
}
#PARA DOIS LOCI
if (loci==2)#Se loci for igual 2, fazer cálculo de diibridismo
{
punnet=matrix (NA,n.gametas,n.gametas, dimnames = list(c("AB", "Ab",
"aB", "ab"), c("AB", "Ab", "aB", "ab")))#Montar matriz com NAs para
preencher com os valores, com número de linhas e colunas de acordo com
número de gametas
for (j in 1:nrow(punnet))
{
for (i in 1:ncol(punnet))
{
punnet[j,i]=ger1a.1[2]+ger1b.1[1]+ger1a.2[1]+ger1b.2[2]#Soma de um
alelo de cada gene do genotipo do progenitor 1(ger1a.1 e ger1a.2) e do
genotipo do progenitor2 (ger1b.1 e ger1b.2)
}
}
#CONVERTER VALORES NUMÉRICOS EM LETRAS PARA O USUARIO
for (j in 1:nrow(punnet)) #Entra no fluxo for com contador j de 1 até
numero de linhas da matriz punnet
{
for (i in 1:ncol(punnet))
{
if (punnet[j,i]==22) #Se a soma de alguma posição for igual a 22
{
punnet[j,i]="AABB"#substitui por "AABB"
}
if (punnet[j,i]==24)#Se a soma de alguma posição for igual a 24
{
punnet[j,i]="AaBB"#substitui por "AaBB"
}
if (punnet[j,i]==36)#Se a soma de alguma posição for igual a 36
{
punnet[j,i]="aaBb"#substitui por "aaBb"
}
if(punnet[j,i]==46)#Se a soma de alguma posição for igual a 46
{
punnet[j,i]="aabb"#substitui por "aabb"
}
if (punnet[j, i]==32)#Se a soma de alguma posição for igual a 32
```

```

    {
      punnet[j,i]="AABb"#substitui por "AABb"
    }
  if (punnet[j,i]==34)#Se a soma de alguma posição for igual a 34
  {
    punnet[j,i]="AaBb"#substitui por "AaBb"
  }
  if (punnet[j,i]==26)#Se a soma de alguma posição for igual a 26
  {
    punnet[j,i]="aaBB"#substitui por "aaBB"
  }
}
}
#SEGUNDA GERAÇÃO
punnet2=matrix (NA,n.gametas,n.gametas, dimnames = list(c("AB", "Ab",
"aB", "ab"), c("AB", "Ab", "aB", "ab")))#Montar matriz com NAs para
preencher com os valores, com número de linhas e colunas de acordo com
número de gametas
for (j in 1:nrow(punnet2)) #Entra no fluxo for com contador j de 1 até
numero de linhas da matriz punnet2
{
  for (i in 1:ncol(punnet2)) #Entra no fluxo for com contador j de 1
até numero de colunas da matriz punnet
  {
    punnet2[j,i]=ger2a.2[j]+ger2b.2[i]#Soma um gameta de um progenitor
com do outro e guarda em y=uma posição da matriz punnet2
  }
}
#CONVERTER VALORES NUMÉRICOS EM LETRAS PARA O USUARIO
for (j in 1:nrow(punnet2)) #Entra no fluxo for com contador j de 1 até
numero de linhas da matriz punnet2
{
  for (i in 1:ncol(punnet2) )#Entra no fluxo for com contador i de 1 até
numero de colunas da matriz punnet2
  {
    if (punnet2[j,i]==22)#Se a soma de alguma posição for igual a 22
    {
      AABB = AABB+1
      punnet2[j,i]="AABB"#substitui por "AABB"
    }
    if (punnet2[j,i]==24)#Se a soma de alguma posição for igual a 24
    {
      AaBB= AaBB+1
      punnet2[j,i]="AaBB"#substitui por "AaBB"
    }
    if (punnet2[j,i]==36)#Se a soma de alguma posição for igual a 36
    {
      aaBb=aaBb+1
      punnet2[j,i]="aaBb" #substitui por "aaBb"
    }
    if(punnet2[j,i]==46)#Se a soma de alguma posição for igual a 46

```

```
{
  aabb=aabb+1
  punnet2[j,i]="aabb"#substitui por "aabb"
}
if (punnet2[j, i]==32)#Se a soma de alguma posição for igual a 32
{
  AABb=AABb+1
  punnet2[j,i]="AABb"#substitui por "AABb"
}
if (punnet2[j,i]==34)#Se a soma de alguma posição for igual a 34
{
  AaBb=AaBb+1
  punnet2[j,i]="AaBb"#substitui por "AaBb"
}
if (punnet2[j,i]==26)#Se a soma de alguma posição for igual a 26
{
  aaBB=aaBB+1
  punnet2[j,i]="aaBB"#substitui por "aaBB"
}
if (punnet2[j,i]==42)#Se a soma de alguma posição for igual a 42
{
  AAAb=AAAb+1
  punnet2[j,i]="AAAb"#substitui por "AAAb"
}
if (punnet2[j,i]==44)#Se a soma de alguma posição for igual a 44
{
  Aabb=Aabb+1
  punnet2[j,i]="Aabb"#substitui por "Aabb"
}
}
}
contador.ab=c(AABB, AAbb, AABb, Aabb,AaBb, AaBB, aaBb, aaBB, aabb)#Cria
vetor que guarda todos os contadores
A.B=AABB+AABb+AaBb+AaBB#Cria objeto que guarda a soma da frequencia de
fenotipos duplamente dominantes
A.bb=Aabb+AAAb#Cria objeto que guarda a soma da frequencia de fenotipos
dominantes para A e recessivos para b
aa.B=aaBb+aaBB#Cria objeto que guarda a soma da frequencia de fenotipos
recessivos para a e dominantes para B
aa.bb=aabb# Cria objeto que guarda a soma da frequencia dos fenotipos
duplamente recessivos
fenotipos=c(A.B, A.bb, aa.B, aa.bb)#Cria vetor que guarda as proporcoes
fenotipicas
fenotipos.m=matrix(fenotipos, 1,4, dimnames = list(c("freq"), c("A_B_",
"A_bb", "aaB_", "aabb")))#Cria objeto para guardar fenotipos como matriz
fenotipos.t=as.table(fenotipos.m)#cria objeto para guardar fenotipos
como tabela
par(mfrow= c(1,2), tcl=0.8, family="serif")#Divide o dispositivo de tela
em duas colunas.
barplot(contador.ab, names.arg=c("AABB", "AAbb", "AABb" , "Aabb","AaBb",
```

```

"AaBB", "aaBb", "aaBB", "aabb"), col="coral2", cex.lab=1.3, pch=19, main =
"Proporção genotípica")#Barplot com as frequencias genotipicas da F2
  balloonplot(fenotipos.t, label =T, show.margins = F, dotcolor =
"coral3", main=" Proporção fenotípica")#balloonplot com as frequencias
fenotipicas
}
  quadros=list(punnet, punnet2) #Cria lista com as matrizes punnet e
punnet2
  names(quadros)=c("Geracao F1", "Geracao F2")#Nome para cada uma das
matrizes
  return(quadros)#Retornar lista com as matrizes
}
if (modelo=="s")#Se o argumento "modelo" for igual a "s", roda a simulacao
{
  if (N==F)#Se o argumento N nao for informado
  {
    stop("Para o argumento modelo='s' deve ser informado N")#interrompe e
exibe a mensagem ao usuario
  }
  filho=c(NA, NA)#Cria objeto "filho", com duas posicoes com NA, para
guardar os valores sorteados
  if (loci==1)#Se loci for igual a 1
  {
    for (i in 1:N)#Entra em um fluxo "for" com contador i de 1 ate N
    {
      filho[1]=sample(ger2a.1,1)#Sorteia um gameta do genotipo ger2a.1(A ou
a)
      filho[2]=sample(ger2b.1,1)#Sorteia um gameta do genotipo ger2b.1(A ou
a)
      if (filho[1]== 1 && filho[2]==1)#Se a posicao 1 de filho for igual a 1
e a posicao 2 for igual a 1
      {
        AA=AA+1#Soma 1 no contador AA
      }
      if (filho[1] != filho[2])#Se a posicao 1 de filho for diferente da
posicao 2
      {
        Aa=Aa+1#Soma 1 no contador Aa
      }
      if (filho[1]==3 && filho[2]==3)#Se a posicao 1 de filho for igual a 3
e a posicao 2 for igual a 3
      {
        aa=aa+1#Soma 1 no contador aa
      }
    }
  }
  contador.a=c(AA,Aa,aa)#Cria vetor para guardar os tres contadores
dominante=AA+Aa#Cria objeto para guardar os contadores de genotipos com
fenotipo dominante
  recessivo=aa#Cria objeto para guardar contador de genotipos com
fenotipo recessivo
  fenotipos=c(dominante, recessivo)#Cria vetor para guardar de acordo com

```

```
os fenotipos
  fenotipos.m=matrix(fenotipos, 1,2, dimnames = list(c("freq"),
c("dominante(A_)", "recessivo(aa)")))#Cria objeto para guardar fenotipo
como matriz
  fenotipos.t=as.table(fenotipos.m)#Cria objeto para guardar fenotipo como
tabela
  par(mfrow= c(1,2), tcl=0.8, family="serif")#Divide o dispositivo de tela
em duas colunas.
  barplot(contador.a, names.arg=c("AA", "Aa", "aa"), col="coral3",
cex.lab=1.3, pch=19, main = "Proporção genotípica")#Barplot com as
frequencias genotipicas da F2 simulada
  balloonplot(fenotipos.t, label =T, show.margins = F, dotcolor =
"coral3", main=" Proporção fenotípica")#balloonplot com as frequencias
fenotipicas simuladas
  esperado.d=0.75*N#Cria objeto para guardar a proporcao esperada de
fenotipos dominantes para o N dado pelo usuario
  esperado.r=0.25*N#Cria objeto para guardar a proporcao esperada de
fenotipos recessivos para o N dado pelo usuario
  chi.valor=sum(((dominante-esperado.d)^2/esperado.d)+((recessivo-
esperado.r)^2/esperado.r))#calculando o chi-quadrado
  alfa=0.05#Cria objeto para o valor de alfa a ser consultado
  df=1#cria objeto com os graus de liberdade para essa simulacao (n-1)
  chi.critico=3.841#Objeto para guardar o valor critico de qui quadrado
encontrado na tabela para o alfa e os df acima
  saida=list(chi.valor,alfa, df, chi.critico)#Cria uma lista para guardar
todos os valores
  valores=as.data.frame(saida, col.names=c("Chi obs","Alfa", "DF", "Qui
crítico"), row.names = "valor")#tranforma a lista em data.frame
  return(valores)#retorna data.frame para o usuario
}
# DOIS LOCI
if (loci==2)#se o argumento "loci" for igual a 2
{
  for (i in 1:N)#Entra em um fluxo for com contador i de 1 ate N
  {
    filho[1]=sample(ger2a.2,1)#Sorteia um gameta do genotipo ger2a.2(AB,
Ab,aB ou ab) e guarda na posicai 1 do objeto "filho"
    filho[2]=sample(ger2b.2,1)#Sorteia um gameta do genotipo ger2b.2(AB,
Ab,aB ou ab)e guarda na posicai 2 do objeto "filho"
    filho3=filho[1]+filho[2]#Soma a posicao 1 e 2 e guarda no objeto
"filho3"
    if (filho3==22)#Se filho3 for igual a 22
    {
      AABB=AABB+1#soma 1 no contador AABB
    }
    if(filho3==42)#Se filho3 for igual a 42
    {
      AAbb=AAbb+1#soma 1 no contador AAbb
    }
    if (filho3==46)#Se filho3 for igual a 46
```

```

{
  aabb=aabb+1#soma 1 no contador aabb
}
if(filho3==26)#Se filho3 for igual a 26
{
  aaBB=aaBB+1#soma 1 no contador aaBB
}
if (filho3==34)#Se filho3 for igual a 34
{
  AaBb=AaBb+1#soma 1 no contador AaBb
}
if (filho3==36)#Se filho3 for igual a 36
{
  aaBb=aaBb+1#soma 1 no contador aaBb
}
if(filho3==32)#Se filho3 for igual a 32
{
  AABb=AABb+1#soma 1 no contador AABb
}
if(filho3==44)#Se filho3 for igual a 44
{
  Aabb=Aabb+1#soma 1 no contador Aabb
}
if(filho3==24)#Se filho3 for igual a 24
{
  AaBB=AaBB+1#soma 1 no contador AaBB
}
}
simulado=c(AABB, AAbb, AABb, Aabb,AaBb, AaBB, aaBb, aaBB, aabb)#Cria
vetor que guarda todos os contadores
A.B=AABB+AABb+AaBb+AaBB#Cria objeto que guarda a soma da frequencia de
fenotipos duplamente dominantes
A.bb=Aabb+AAbb#Cria objeto que guarda a soma da frequencia de fenotipos
dominantes para A e recessivos para b
aa.B=aaBb+aaBB#Cria objeto que guarda a soma da frequencia de fenotipos
recessivos para a e dominantes para B
aa.bb=aabb# Cria objeto que guarda a soma da frequencia dos fenotipos
duplamente recessivos
fenotipos=c(A.B, A.bb, aa.B, aa.bb)#Cria vetor para guardar todas as
proporcoes fenotipicas
fenotipos.m=matrix(fenotipos, 1,4, dimnames = list(c("freq"), c("A_B_",
"A_bb", "aaB_", "aabb") ))#Cria objeto para guardar fenotipos em uma matriz
fenotipos.t=as.table(fenotipos.m)##Cria objeto para guardar fenotipos em
uma tabela
par(mfrow= c(1,2), tcl=0.8, family="serif")#Divide o dispositivo de tela
em duas colunas.
barplot(simulado, names.arg=c("AABB", "AAbb", "AABb", "Aabb","AaBb",
"AaBB", "aaBb","aaBB", "aabb"), col="coral2", cex.lab=1.3, pch=19, main =
"Proporção genotípica simulada)#Barplot com frequencias genotipicas
simuladas
balloonplot(fenotipos.t, label =T, show.margins = F, dotcolor =

```

```
"coral3", main=" Proporção fenotípica simulada")#Balloonplot com frequencias
fenotipicas simuladas
esperado.A.bb=0.1825*N#Cria objeto para guardar a proporcao esperada de
fenotipo A_bb para o N dado pelo usuario
esperado.A.B=0.5625*N#Cria objeto para guardar a proporcao esperada de
fenotipo A_B_ para o N dado pelo usuario
esperado.aa.B=0.1825*N#Cria objeto para guardar a proporcao esperada de
fenotipo aaB_ para o N dado pelo usuario
esperado.aa.bb=0.0625*N#Cria objeto para guardar a proporcao esperada de
fenotipo aabb para o N dado pelo usuario
chi.valor=sum((((A.B-esperado.A.B)^2)/esperado.A.B)+(((A.bb-
esperado.A.bb)^2)/esperado.A.bb)+(((aa.bb-
esperado.aa.bb)^2)/esperado.aa.bb)+(((aa.B-
esperado.aa.B)^2)/esperado.aa.B))#calculando o chi-quadrado
alfa=0.05#Cria objeto para o valor de alfa a ser consultado
df=3#cria objeto com os graus de liberdade para essa simulacao (n-1)
chi.critico=7.815#objeto para guardar o valor critico de qui quadrado
encontrado na tabela para o alfa e os df acima
saida=list(chi.valor,alfa, df, chi.critico)#Cria uma lista para guardar
todos os valores
valores=as.data.frame(saida, col.names=c("Qhi obs","Alfa", "DF", "Qui
crítico"), row.names = "valor")#tranforma a lista em data.frame
return(valores)#Retorna o data.frame "valores"
}
}
}
```

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2018:alunos:trabalho_final:camila.avbio:mendel



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