Tiago da Silva Ribeiro

Mestrando em Genética no IB-USP. Realiza pesquisa na área de filogeografia, com foco em aves da Mata Atlântica.

exec

Proposta de Projeto

Plano A

Agrupando Localidades por Mapas de Probabilidade de Distribuição

Utilizando recursos de Modelagem de Nicho Ecológico é possível inferir a probabilidade de ocorrência de determinada espécie com base em um conjunto de dados de ocorrência, e em alguns casos de ausência também, da mesma. A inferência também pode ser deslocada no tempo ou no espaço.

A função aqui proposta usará como arquivos de entrada: um mapa com probabilidade de ocorrência para cada posição, um arquivo com localidades referênciadas com coordenadas geográficas, um arquivo fasta com alinhamento de sequências de DNA de um número qualquer de indivíduos e um arquivo que relacione cada indivíduo a uma localidade. (o número de arquivos pode ser modificado, mas as informações de entrada necessária serão estas)

A função criará um objeto intermediário, que opcionalmente poderá ser salvo em um arquivo, que contém as mesmas localidades do arquivo de entrada, porém separadas em dois grupos, um grupo com as localidades que coincidem com as localidades com determinada de probabilidade de ocorrência ou maior, e um grupo com as localidades que coincidem com posições menores do que a probabilidade de ocorrência especificada.

A partir disso, temos como argumento da função um número que se refira à probabilidade que deve ser utilizada como limiar para separação dos grupos.

Em segundo momento, a função analisará a diversidade genética de cada um desses grupos e efetuará comparações entre os dois. Para algumas das análises de diversidade genética realizadas serão efetuados testes de significancia, com o objetivo de verificar se os valores são significativamente diferentes.

Temos então os argumentos “opcionais” da função, que correspondem aos testes de diversidade genética, que poderão ser “ligados” ou “desligados”. Os testes implementados serão utilizados chamando funções já existentes de outros pacotes, não se pretende alterá-las.

Teremos assim como arquivo de saída uma lista de valores de diversidade genética de cada grupo.

Plano B

Plano B *- opção A reduzida* Uma versão do Plano A que tem o arquivo intermediário como ponto final da função, deixando as análises de diversidade genética ausentes.

***Plano B** - B - Guia de Estudos*

Um Plano B original, independente do Plano A.

Como arquivo de entrada teremos um arquivo com 4 colunas:

na primeira: os tópicos que precisam ser estudados,

na segunda: o número de páginas de cada tópico (caso as páginas tenham tamanhos diferentes, será preciso normalizar o número antes de adicionar os dados),

na terceira: um índice de 0 a 1 representando o grau de dificuldade em relação ao estudo do tópico, sendo 0 o menor grau, e 1 o maior,

a quarta coluna: é de preenchimento opcional, também variando de 0 a 1. Deverá conter a proporção esperada de que aquele tópico caía na prova, normalmente uma estimativa do aluno. Caso deixada em branco (NA), cada tópico terá a mesma probabilidade de ocorrência na prova. Caso algum valor seja atribuído: as linhas com valores preenchidos seguirão definição do usuário, e as demais obterão uma divisão igualitária do restante da probabilidade. Se o valor da linha for 0, o tópico só será indicado para estudo em caso de tempo extra. Valores negativos ou que soma total de valores maior do que 1 retornaram erro.

A função terá como argumento o tempo de estudo disponível até a hora da prova e a nota que o aluno almeja tirar.

A função retornará o que deve ser estudado e em que ordem. Em decorrência do tempo disponível (pouco tempo para estudo), poderá ser sugerido que gaste parte dele procurando por resumo com os colegas. Em decorrência do tempo disponível (pouquíssimo tempo para estudo), a função dará mensagem de alerta, avisando o usuário de que ele não deveria ter perdido tempo calculando isso no R.

Comentários das propostas (Leo)

Assumo que você já sabe gerar os mapas com probabilidades de ocorrência, e estes mapas podem ser importados para o R em um formato adequado para realizar as análises. Caso ainda não domine esta operação, será uma barreira difícil (mas não impossível) para gerar a função. Em todo caso acho que a proposta A simplificada, sem os dados genéticos, seja melhor para o trabalho final. A inclusão da informação genética pode ser deixada para depois. Facilitaria o entendimento (e seu trabalho) se tu pudesse decompôr a proposta da função em tarefas menores, passo-a-passo.

Algumas dúvidas: E se o mapa apresenta manchas de maior probabilidade de ocorrência, não seria mais interessante comparar os indivíduos em diferentes manchas? Ou existe alguma hipótese a priori que exija uma comparação entre indivíduos em habitats diferentes com diferentes probabilidades de ocorrência (um proxy de qualidade do habitat?)? Por que não plotar diretamente os indivíduos no mapa através de suas coordenadas geográficas? Se o grid com a probabilidade de ocorrência for uma superfície raster, tu pode atribuir um valor para cada indivíduo plotado no mapa.

Enfim, acho a proposta interessante mas alguns pontos não ficaram claros. Uma clarificação destes pontos é necessária para que a exequibilidade da função dentro do prazo seja avaliada.

Respostas

Obrigado pelos comentários, vou tentar clarificar um pouco, então.

Eu consigo fazer os mapas, atualmente eu uso o MaxEnt, mas pretendo passar a utilizar o pacote BIOMOD, do R. Entretanto não conheço a estrutura dos arquivos grid, preciso estudá-la para entender como executar a função.

Os mapas podem sim ter manchas de maior probabilidade, não havia pensado nisso. Talvez seja possível diferir um mapa com manchas de um com distribuição contínua utilizando comparações condicionais, mas não por hora não consigo pensar me como fazê-lo, principalmente por não conhecer a estrutura dos arquivos.

O que eu estava fazendo para conseguir separar as minhas localidades em dois grupos era justamente plotar os pontos no mapa, mas o trabalho me parecia pouco preciso, e com um volume grande de pontos, ou de espécies, ou de mapas gerados, pode ficar cada vez mais trabalhoso e repetitivo. Assumo que a falta de precisão que eu encontrei ao fazer essa separação a olho foi o motivo crucial para a proposta da função. Com essa função eu também poderia escolher, mais facilmente, diferentes probabilidades limiares, e poderia assim testar mais simplesmente modelos mais conservadores (com limiares altos) em relação à resposta de modelos mais relaxados (limiares baixos).

A idéia de atribuir o valor de probalidade no grid específico à espécie me parece uma estratégia muito boa para realizar a função, mas ainda seria preciso separar os grupos entre as diferentes manchas, caso ocorram.

Aceito a sugestão de tornar a proposta A reduzida a proposta principal. Gerando assim como arquivo de saída um arquivo com uma separação dos pontos de localidades. A parte deixada de fora pode ser uma função futura, utilizando como arquivo de entrada o arquivo de saída gerado nessa função.

Tentarei quebrar a função em diferentes tarefas e as postarei aqui novamente.

Novamente agradeço os comentários.

Função - Guia de Estudos - Proposta B

Código da Função

```
guia.de.estudos <- function(planilha, tempo.disponivel, nota)
{
  conta.na = 0
  soma.proporcao = 0
  for(busca.na in 1:length(planilha[,4]))
  {
    if(is.na(planilha[busca.na,4]))
    {
      conta.na = conta.na + 1
    }
```

```
    else
    {
      soma.proporcao = soma.proporcao + planilha[busca.na,4]
    }
  }
  if(soma.proporcao>1|soma.proporcao<0)
  {
    return(cat("Erro!! Corrija os valores introduzidos como proporção dos
conteúdos na prova! A soma deles deve estar entre 0 e 1. ;)"))
  }
  if(nota>10 | nota<0)
  {
    return(cat("Erro!! A nota almejada tem que estar entre 0 e 10. ;)"))
  }
  nota = nota/10
  adiciona.proporcao = (1-soma.proporcao)/conta.na
  for(busca.na in 1:length(planilha[,4]))
  {
    if(is.na(planilha[busca.na,4]))
    {
      planilha[busca.na,4] = adiciona.proporcao
    }
  }
  planilha$pag.norm.dificuldade = planilha[,2]*(planilha[,3]+1) #correção do
número de páginas em relação à dificuldade de leitura das mesmas
  planilha$tempo.necessario = round(planilha$pag.norm.dificuldade * 2.1)
  round(planilha$tempo.necessario)
  planilha$indice.prioridade = planilha[,4]/planilha$tempo.necessario
#indice de prioridade para guia de estudos
  indice.ordenado = order(planilha$indice.prioridade, decreasing=TRUE)
  planilha$indice.crescente = 0
  for(prioridade in 1:length(planilha$indice.crescente)) ##Prioridade
ordenada crescentemente (tópicos com menor valor do indice são prioritários
sobre tópicos com valores maiores)
  {
    planilha[indice.ordenado[prioridade], 8] = prioridade
  }
  tempo.disponivel.round = round(tempo.disponivel)
  contador.tempo.necessario = planilha$tempo.necessario
  planilha[, 9] = 0
  contador.nota = 0
    for(i in 1:length(planilha[,7]))
    {
      while(contador.tempo.necessario[planilha$indice.crescente==i] > 0 &
tempo.disponivel.round>0 & contador.nota<nota)
      {
        contador.tempo.necessario[planilha$indice.crescente==i] =
contador.tempo.necessario[planilha$indice.crescente==i] -1
        tempo.disponivel.round = tempo.disponivel.round - 1
        planilha[planilha$indice.crescente==i, 9] =
```

```
planilha[planilha$indice.crescente==i, 9] + 1
        contador.nota = contador.nota +
planilha[planilha$indice.crescente==i,7]
      }
    }
  if(tempo.disponivel.round>0)
  {
  planilha[,10] = 0
  while(tempo.disponivel.round>0)
  {
    contador.tempo.necessario = planilha$tempo.necessario
    for(i in 1:length(planilha[,7]))
    {
      while(contador.tempo.necessario[planilha$indice.crescente==i] > 0 &
tempo.disponivel.round>0)
      {
        contador.tempo.necessario[planilha$indice.crescente==i] =
contador.tempo.necessario[planilha$indice.crescente==i] -1
        tempo.disponivel.round = tempo.disponivel.round - 1
        planilha[planilha$indice.crescente==i, 10] =
planilha[planilha$indice.crescente==i, 10] + 1
        contador.nota = contador.nota +
planilha[planilha$indice.crescente==i,7]
      }
    }
  }
  }
  cat(paste("Para tentar tirar nota", nota*10, "Recomedo que estude na
seguinte ordem: ", "\n", "\n"), sep=" ")
  for(i in 1:length(planilha[,1]))
  {
    cat(paste(planilha[planilha$indice.crescente==i,1], "por",
planilha[planilha$indice.crescente==i, 9], "minutos.", "\n", sep=" "))
  }
  if(length(planilha[1,])==9)
  {
    cat("\n", "Fim!", "\n", "\n")
    return(data.frame(Assunto = planilha[,1], Minutos.Estudo.Minimo =
planilha$V9))
  }
  if(length(planilha[1,]==10))
  {
    cat("\n", "Seguindo o roteiro acima é espero que tenha estudado o
suficiente para tirar", nota*10, ". Como ainda terás tempo extra, se quiser
estudar mais, recomendo que divida o seu tempo assim:", "\n", "\n")
    for(i in 1:length(planilha[,1]))
    {
      cat(paste(planilha[planilha$indice.crescente==i,1], "por mais",
planilha[planilha$indice.crescente==i, 10], "minutos.", "\n", sep=" "))
    }
    cat("\n", "Fim!", "\n", "\n")
```

```
    return(data.frame(Assunto = planilha[,1], Minutos.Estudo.Minimo =
planilha$V9, Minutos.Estudo.Extra = planilha$V10))
  }
}
```

Página de Ajuda

```
guia.de.estudos               package: tsr                R Documentation


Description:

     Função para organizar o tempo disponível de estudo para uma prova.
     O cálculo é realizado com uma velocidade média de leitura de 1000
caracteres por minuto, valor obtido com base em um arredondamento da
velocidade média de leitura de indivíduos saudáveis de idade entre 19 e 35
anos (Messias et al. 2008).

Usage:

     guia.de.estudo(planilha, tempo.disponivel, nota)


Arguments:

     planilha       um data.frame contendo na primeira coluna o nome dos
tópicos a serem estudos, na segunda o número de páginas, na terceira a
dificuldade relativa com a leitura de cada tópico e na quarta a proporção
esperada do tópico na prova (proporção final deve estar entre 0 e 1).
     tempo.disponivel   o tempo disponível para estudo, em minutos.
     nota       a nota almejada na prova, entre 0 e 10.


Details:

     Ao inserir os dados na planilha, o número de páginas deve ser
padronizado em relação a uma página A4 com fonte arial 12, o equivalente a
30 linhas por página, com uma média de 70 toques por linha e espaçamento
duplo.
     Os valores de dificuldade relativa equivalem à porcentagem de
dificuldade com a leitura sobre o tema, 0% (zero) significa nenhuma
dificuldade com a leitura, 100% significa 100 porcento de dificuldade com o
tema, a dificuldade pode ser negativa ou superior a 100%. Os valores devem
ser passados em formato decimal (exemplo: 100% = 1, 50% = 0,5)
     Caso não queira informar a proporção esperada no tópico da prova,
deixar em branco e NÃO preencher com 0 (zero).
```

```
Value:

     A função retorna de forma visual a ordem em que os tópicos devem ser
estudados e o tempo de estudo que deve ser dedicado a cada tópico.
     Em caso de tempo extra, a função também retorna uma sugestão de divisão
do tempo extra.
     Além da informação visual a função retorna um data.frame que contem os
tópicos que devem ser estudos na primeira coluna, a divisão do tempo de
estudo minimo para cada tópico na segunda coluna e a sugestão de divisão do
tempo extra de estudo na terceira coluna. O data.frame não está ordenado na
sequência de estudo sugerida pela saída visual da função.


Warning:

     Em textos com figuras e tabelas, e em textos que estão em página A4 com
30 linhas e uma média de 70 caracteres por linha o usuário deve fazer uma
estimativa de conversão do número de páginas real para o número de páginas
ideal especificado acima, que não considera a existência de imagens e
tabelas.
     Caso a segunda ou a terceira coluna contenham valores nulos ou que não
sejam numéricos, a função não funcionará.


Note:

     Um tópico cuja proporção esperada na prova é de 0 (zero) será
descontado do tempo mínimo de estudo.


Author(s):

     Tiago da Silva Ribeiro


References:

     Messias, A.; Cruz, A. A. V.; Schallenmüller, S. J. & Trauzettel-
Klosinski, S. 2008. Textos padronizados em português (BR) para medida da
velocidade de leitura - comparação com quatro idiomas europeus. Arq Bras
Oftalmol. 71(4):553-8.


Examples:

     # baixe o arquivo chamado "topicos.csv", este arquivo contém três
tópicos para estudo, com número de páginas, dificuldade relativa e proporção
esperada na prova.
     dados <- read.csv("topicos.csv", header=TRUE, sep=";", dec=",")
     guia.de.estudos(dados, 120, 8)  #exemplo sem tempo extra
     guia.de.estudos(dados, 200, 10) #exemplo com tempo extra
```

Arquivos Para Download

Código da Função Guia de Estudos

Arquivo em formato csv para ser utilizado com o exemplo de ajuda

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

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:alunos2012:alunos:trabalho_final:tsr:start

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