Júlia Rodrigues Barreto

Doutoranda em Ecologia (IB-USP) sob orientação da Prof Liz Nichols. Meu projeto de tese pretende investigar como a perda de floresta afeta interações interespecíficas.

Meus Exercícios

Propostas de Trabalho Final:

Proposta A – Estimativa área foliar afetada por herbivoria (Rbivory)

Contextualização/Objetivo

Uma das maiores barreiras metodológicas no estudo de herbivoria foliar é atingir a precisão na estimativa de área afetada. O Maior fator de imprecisão é o fato de que muitos dos danos se localizam nas bordas das folhas dificultando a inferência de área original mesmo em software consolidado nesse objetivo (e.g. ImageJ). As duas imagens abaixo esquematizam como as folhas vêm escaneadas do campo para serem tratadas e terem área medida, e como elas idealmente deveriam ser reconhecidas para medição de área original (com contorno real). A presente proposta visa produzir uma função para recompor borda original, calcular a área (cm2) removida de folhas baseada no tamanho médio da folha e número de partículas presentes na imagem fornecida.

»»»

O caminho que se pretende seguir para alcançar tal objetivo é utilizar de ferramentas de Sistemas de Informação Geográficas (SIG) no R para que se possa obter informações de forma e tamanho (e.g. razão comprimento-largura como na imagem abaixo, e/ou perímetro-área) e assim prosseguir na medição.

Possíveis usos da função

A função pode ser usada para estimar tanto área quanto proporção atingida por herbivoria. Inova por permitir inferir área original recompondo contorno mesmo quando o dano está presente nas bordas.

Planejamento da Função

Entrada: Rbivory (n, spp, area= T)

• Imagem binária (apenas folhas em preto e fundo branco) em um dos formatos compatíveis (.png ou raster);

• n = número de partículas/folhas presentes na imagem (classe: integer, N > 0);

• spp = nome científico da planta (classe: character);

• Se o interesse for medir área e não apenas proporção removida (area= TRUE), essa imagem deve ter escala conhecida;

Verificação de Parâmetros

• Imagem é suportada e compatível? Senão retorna “Imagem de tamanho ou formato incompatível”;

• N é número integer >0? Senão retorna “n precisa ser inteiro e >0”;

• Nome da espécie consta no banco de dados? Senão retorna “espécie não catalogada”;

• Se ‘area= T’, dpi deve ser fornecido. Senão retorna “Na ausência de valor de escala fornecido, apenas proporção será calculada”.

Pseudo-código

a) Transformar imagem em raster;

b) Calcular razão comprimento/largura de cada partícula foliar na imagem;

c) Baseado nessa razão por folha, traçar em cada uma delas um segmento ligando as bordas desconectadas

d) Medir por folha número de pixel brancos e pretos

e) Com informação de área por píxel, converter pixel preto por folha em área remanescente e a soma de brancos e pretos em área total da folha

f) Calcular essa razão para extrair a proporção

g) Unir cada um desses vetores gerados num só data.frame resultado

Saída

• Data frame contendo código de cada partícula medida por imagem com valores de área total da folha, área removida e proporção (prop: class numeric $0 \leq prop \leq 1$);

• Imagem recomposta com o feixe convexo na região da borda faltante.

Dificuldades

• Entender melhor as possibilidades de ferramentas e se através do SIG é mesmo o melhor caminho;

• Se a entrada da imagem e transformação em raster não compromete a individualização das partículas (para retornar informação por folha);

• Não tenho certeza se possível retornar imagem com os segmentos refeitos;

• Em último caso posso utilizar de banco de dados com área média de folha já definida por espécie e fazer a medição de área perdida folha a folha mas isso perde um pouco o propósito além de ficar mais impreciso já que se basearia em média.

Proposta B – Roteiro de Turismo (touR)

Contextualização/Objetivo

Muitas vezes lidamos com dificuldades no planejamento de viagem sendo ainda mais complicado conciliar orçamento e tempo hábil para turismo. Propõe-se a função 'touR' que seja capaz de simular roteiro de viagem com pontos turísticos principais na cidade de interesse condicionado ao tempo de permanência.

Possíveis usos da função

A função pode ser usada para recomendação de principais pontos turísticos para cidades presentes no banco de dados.

Planejamento da Função

Entrada touR(cit, d, bud= T) ← data.frame ("dados.tour.csv")

a) cit= nome cidade (classe: character);

b) d= tempo de permanência em dias (classe: integer, >0);

c) bud= indicar limitação no orçamento (e.g. 'bud=T' para priorizar turismo de baixo custo)

Verificação de Parâmetros

• Nome da cidade consta no banco de dados? Senão retorna "cidade ainda não catalogada";

• n é número inteiro maior que zero? Senão retorna "n precisa ser inteiro e >0";

• Se para o tempo de permanência estabelecido na cidade de interesse não for possível combinar opções de baixo custo retornar "Cidade de alto custo, apenas turismo patrão".

Pseudo-código

a) Função começa com uma mensagem de interação com o usuário e em seguida baixa os dados de página online;

b) Tratando o input de cidade para que a forma que foi escrita se adeque a do banco de dados da funcao;

b.1 criando lista dos acentos possiveis para remover e colocar em minusculo

b.2 colocando td minusculo e removendo acentos em ciclo para caso haja + de um

b.3 substituir espacos por underscores pq esta assim no dataset exemplo

b.4 apos de tratado achar se ha match com cidades dos dados

c) Verificação de parametros

c.1 conferir se contem a cidade selecionada no banco de dado

1. cria subset filtrando do df original apenas dados da cidade selecinada

c.2 argumento dias tem que ser numero positivo e inteiro

c.3 confere se a permanência é mais longa que nosso banco de dados pode suprir

c.4 Confere se houve input para o argumento budget

1. se TRUE, conferir se ha atracoes gratuitas suficientes no destino para o n de dias (metade dos passeios)

d) Gerar roteiro para orcamento limitado (if budget == TRUE)

e) if budget = FALSE: gerar roteiro com as principais atrações sem condição a proporção de atividades gratuitas

f) Retorna o data.frame 'roteiro'

Saída

```
Data.frame 'roteiro' com quatro opções de passeio por dia de permanência
```

Dificuldades

- Gerar e incluir na função um data.frame catálogo contendo: cidades, atrações ordenadas por importância e gratuidade (0 ou 1).

Comentários sobre as Propostas

Olá, Júlia. Gostei bastante do intuito da sua proposta A, porém ela me parece bem ambiciosa. Não acompanhei você de perto durante as aulas então não sei se você está subestimando a tarefa. A proposta B também é bem legal, com alguns ajustes pode ser bem interessante de ser realizada. O que posso comentar mais especificamente sobre as propostas é:

Proposta A:

1- Você tem alguma ideia de como “completar” as bordas das folhas? O seu algoritmo passa meio por cima nessa questão, a qual me parece fundamental para o desenvolvimento da função.

2- Não entendi o papel dos argumentos 'n' e 'spp' no funcionamento da sua função.

3- Se “area = T” faz necessário que seja fornecido o dpi, então o dpi também é um argumento.

Proposta B:

1- Eu utilizaria o argumento budget como um numeric onde a pessoa coloca o valor que ela quer gastar, então a função sugere uma gama de rolês plausíveis com esse orçamento.

2- Você provavelmente terá de lidar com problemas do tipo: Seu banco tem a cidade de “SAO PAULO” e a pessoa procura por um rolê em “São Paulo”. Trabalhar com essas strings pode dar um trabalho razoável. Sugiro que ou você trabalhe com uma interação através do console para guiar a seleção das cidades.

Achei ambas as propostas interessante. Eu prefiro a proposta A por ter um objetivo mais bem delineado e ser mais geral dentro do seu campo, porém conseguir automatizar todo este processo parece um trabalho bastante difícil, principalmente a parte de completar a borda da folha automaticamente. Tente desenhar um algoritmo/pseudocódigo mais minucioso para que eu possa ter uma noção melhor do tamanho do trabalho. Sugiro que você pense em uma maneira de contornar este passo do desenho da borda para levar a cabo sua função. Fazer com que a pessoa entregue o desenho da borda completado na própria folha é um trabalho muito surreal?

Qualquer coisa se quiser mandar conversar sobre a função manda email pra daniel.magalhaes.lima@usp.br.

Até!

Daniel

FUNÇÃO touR

Help!

```
touR                              package:unknown                      R
Documentation
Função desenvolvida para montar um roteiro de turismo selecionando pontos
turísticos de uma cidade indicada.

Description:

touR é uma função interativa que seleciona as principais atividades
turísticas para uma cidade selecionada a partir de um data.frame e organiza
estes elementos em uma matriz indexada pelos dias de viagem, formando um
roteiro de quatro atividades para cada dia. A seleção dos elementos do
data.frame é feito a pa    rtir de uma indexação destes elementos pela
posição no dataframe, número de dias desejado e se há limitação no
orçamento.

Usage:

touR(cit, d, bud) ←data.frame

Arguments:

cit     cidade para a qual se planeja a viagem (classe ‘character’)

d   tempo de permanência em dias (classe ‘numeric’, inteiro e positivo)

bud prioridade de turismo de baixo custo (classe ‘logic’, TRUE or FALSE)

Details:
```

A função ‘touR’ é capaz de simular um roteiro de viagem com os principais pontos turísticos na cidade de interesse adequado ao tempo de permanência. Cidade (cit) e número de dias (d) são argumentos de entrada mandatórios, budget não tem um default porém a função consulta o usuário caso não seja preenchido. Para melhor garantir que a cidade procurada seja encontrada no nosso banco de dados, recomenda-se que o nome seja escrito em inglês, letras minúsculas e susttuindo espaços por underline (“_”). O banco de dados que vem acompanhado da função contém 51 opções de cidades que podem ser indicadas como destino, bem como seus principais pontos turísticos em ordem de importância e se são atrações que requerem investimento ou gratuitas.

Value:

A função retorna um data.frame chamado roteiro com colunas para cada dia e quatro opções de passeio por dia nas linhas.

Warning:

Caso os argumentos de entrada não cumpram os requisitos, a função é interrompida na fase de verificação. A função será interrompida nos seguintes casos: (i) a cidade indicada não conste no banco de dados ou (ii) o número de dias (d) indicado não seja número inteiro e/ou seja menor que zero..

Notes:

As cidades disponíveis no banco de dados da versão atual são:

```
 [1]"johannesburgh"  "berlin"         "toronto"        "mumbai"
"madrid"
 [6] "munich"         "dublin"         "rome"           "los_angeles"
"miami"
[11] "prague"         "vienna"         "shanghai"       "taipei"
"osaka"
[16] "milan"          "amsterdam"      "aberdeen"       "edinburgh"
"glasgow"
[21] "barcelona"      "istanbul"       "hong_kong"      "new_york"
"kuala_lumpur"
[26] "seoul"          "tokyo"          "singapore"      "dubai"
"paris"
[31] "london"         "bangkok"        "rio_de_janeiro" "sao_paulo"
"foz_do_iguacu"
[36] "salvador"       "brasilia"       "porto_alegre"   "manaus"
"florianopolis"
[41] "fortaleza"      "belo_horizonte" "curitiba"       "natal"
"recife"
[46] "buenos_aires"   "santiago"       "bogota"         "lima"
"caracas"
[51] "quito"
```

```
Author(s):

Júlia Rodrigues Barreto
e-mail: barretoju91@gmail.com

References:

Todas as informações que compõem o banco de dados foram extraídas do site de
viagens ‘Tripadvidor.com‘.
A função faz uso dos seguintes pacotes: ‘data.table’, ‘stringdist’, ‘dplyr’,
Que devem estar instalados para que ela seja executada devidamente.

Examples:

## Teste cidade de São Paulo, 2 dias (atrações suficientes) e budget= FALSE.
touR("sao_paulo", 2, bud= F)

## Exemplo com erro de digitação no nome da cidade
touR("São Paolo", 2, bud=T)

## Exemplo em que o número de dias excede a quantidade de atrações
disponíveis nos dados
touR("São Paulo", 4, bud= T)

## Exemplo com argumento budget ausente
touR("São Paulo", 2)
```

Código

```
touR <- function(cit, d, bud){
  ## mensagem interacao com user
  cat("Vamos planejar sua viagem!")
  #### 0 Getting data from online source: the R course wiki page ####
  require(data.table)
  data <-
fread('http://ecologia.ib.usp.br/bie5782/lib/exe/fetch.php?media=bie5782:01_
curso_atual:alunos:trabalho_final:barretoju91:dados.tour.csv')
  #### 1 tratando a cidade de entrada da forma que foi escrita na funcao
####
  #1.1 criando lista dos acentos possiveis para remover e colocar em
minusculo
  unwanted <- list('Š'='s', 'š'='s', 'Ž'='z', 'ž'='z', 'À'='a', 'Á'='a',
'Â'='a', 'Ã'='a', 'Ä'='a', 'Å'='a', 'Æ'='a', 'Ç'='c', 'È'='e',
'É'='e','Ê'='e', 'Ë'='e', 'Ì'='i', 'Í'='i', 'Î'='i', 'Ï'='i', 'Ñ'='n',
'Ò'='o', 'Ó'='o', 'Ô'='o', 'Õ'='o', 'Ö'='o', 'Ø'='o', 'Ù'='u', 'Ú'='u',
'Û'='u', 'Ü'='u', 'Ý'='y', 'Þ'='b', 'ß'='ss', 'à'='a', 'á'='a', 'â'='a',
'ã'='a', 'ä'='a', 'å'='a', 'æ'='a', 'ç'='c', 'è'='e', 'é'='e', 'ê'='e',
'ë'='e', 'ì'='i', 'í'='i', 'î'='i', 'ï'='i', 'ð'='o', 'ñ'='n', 'ò'='o',
'ó'='o', 'ô'='o', 'õ'='o', 'ö'='o', 'ø'='o', 'ù'='u', 'ú'='u', 'û'='u',
```

```
'ý'='y', 'ý'='y', 'þ'='b', 'ÿ'='y' )
  #1.2 colocando td minusculo e removendo acentos em ciclo para caso haja +
de um
  for(i in seq_along(unwanted)){
    cit <- tolower(gsub(names(unwanted)[i],unwanted[i],cit))  }
  #1.3 substituir espacos por underscores pq esta assim no dataset exemplo
  cit <-  gsub('([[:punct:]])|\\s+','_',cit)
  #1.4 apos de tratado achar se ha match com cidades dos dados
  require(stringdist)
  cit <- data$city[amatch(cit,data$city,maxDist = 3)]
  #### 2 Verificando parametros ####
  #2.1 conferir se contem a cidade selecionada no banco de dado
  if(!cit %in% data$city) {
    stop("Infelizmente a cidade selecionada não consta nas opções de destino
do banco de dados.") }       # funcao interrompida e avisa o motivo ao
usuario
  #### subset filtrando do df original apenas dados da cidade selecinada
####
  require(dplyr)
  data.cit <- data %>% filter(data$city==cit)
  #2.2 argumento dias tem que ser numero positivo e inteiro
  if (d != round (d) | d <= 0){
    stop ("Numero de dias deve ser um numero inteiro e > 0!")  }
  #2.3 permanência mais longa que nosso banco de dados pode suprir? Aviso!
  if (d > floor(length(which(data$city== cit))/4)){ ## setamos 4
atracoes/dia como padrao
    message("Seu tempo de permanência nessa cidade excede a quantidade de
principais atrações para esse destino no nosso banco de dados.")
    ## 2.3.1 pergunta ao usuario com duas opcoes sobre como proceder
    quest<- menu(c("1", "2"), title="Você gostaria de (1) repensar os planos
para otimizar seu tempo com atividades ou (2) Manter número de dias com
todas as atrações possíveis ao longo da estadia mesmo assim?")
    ### opcao 1- se decidir reduzir o numero de dias
    if (quest == "1") {
      ## consultar qual o novo tempo de permanencia dando opcao adequada ao
viavel
      d2 <- menu(1:floor(length(which(data$city == cit))/4),title= "Já que
escolheu por repensar, indicar novo tempo de permanência em dias desejado?")
      ## confere parametro dia atualizado: inteiro, >0 e se as atividades
sao suf
      if (d2 != round(d2) | d2 <=0){
        stop("Seu novo número de dias deve ser inteiro, >0 e menor que o
anterior!")}
      ## se d2 atende aos parametros vamos seguir com ele como d
      d <- d2
    }
    ### opcao 2, sem limites mesmo
    if (quest==2) {
      roteiro <<- matrix(data.cit$tour[1:(d*4)], nrow= 4, ncol= d, dimnames
= list(paste0("Passeio ",1:4), paste0("Dia ", 1:d))) }
```

```
  }
  #### 3) traveling on a budget? ####
  #3.0 se deixou de preencher o argumento budget
  if (missing(bud)){ ## nao rolou
    message("Percebemos que deixou de preencher o argumento budget.")
    bud <- menu(c("Sim", "Não"), title="Sua viagem está sendo planejada com
orçamento limitado?") ## pergunta pra saber qual proporcao das atracoes tem
que ser gratuita
    if (bud == 1) {bud <- TRUE}
    if (bud==2){ bud<- FALSE}
  }
  #3.1 preencheu budget = TRUE
  if (bud== TRUE){
    ### conferir se ha atracoes gratuitas suficientes no destino para o n de
dias
    ## metade menor inteira das atividades eh 0800? Se nao for, stop
    if((floor(length(data.cit$paid[data.cit$paid==0])))< (d*4/2)){
      ## interrompe por nao ter atracoes gratuitas suficientes para
preencher metade do roteiro de viagem
      stop("Infelizmente dentre as atrações principais não há opções
gratuitas suficientes para esse destino. Por favor, reconsidere limitação no
orçamento, por exemplo, diminuindo o tempo de permanência.") }
    ####  orcamento limitado ####
    # vetor atracoes pagas
    pg <- data.cit$tour[data.cit$paid==1]
    pg <- pg[1:(d*4/2)]
    # vetor coisas gratis
    gratis <- data.cit$tour[data.cit$paid==0]
    gratis <- gratis[1:(d*4/2)]
    # junta para criar matriz e sorteia ordem
    roteiro <<- matrix(sample(c(pg,gratis)),nrow= 4, ncol= d, dimnames =
list(paste0("Passeio ",1:4), paste0("Dia ", 1:d)))
  }
  ## 3.2 preencheu que nao ha limites argumento budget = False
  if (bud== FALSE){
    roteiro <<- matrix(data.cit$tour[1:(d*4)], nrow= 4, ncol= d, dimnames =
list(paste0("Passeio ",1:4), paste0("Dia ", 1:d)))
  }
  return(roteiro)
}
```

Dados:dados.tour.csv

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

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2018:alunos:trabalho_final:barretoju91:start

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