

Sheina Koffler



Mestranda do Departamento de Ecologia, Instituto de Biociências da USP. Laboratório de Abelhas

Projeto de Pesquisa: Influência da quantidade de recursos alimentares e da sazonalidade sobre a produção de sexuais em *Scaptotrigona aff. depilis* (Apidae, Meliponini)

Exercícios Resolvidos

[ex_1.r](#) [ex_2.r](#) [ex_3.r](#) [ex_4.r](#) [ex_5.r](#) [ex_6f.r](#) [ex_7.r](#) [ex_9.r](#)

Trabalho Final

Plano A

Favos são estruturas em formato de disco, compostos por células hexagonais justapostas, nas quais se desenvolvem as crias das abelhas. No caso da tribo Meliponini (“abelhas sem ferrão”), os favos de cria geralmente tem uma aparência circular, porém, com bordas irregulares, o que é atribuído a um padrão ao acaso da construção das células (Eterovic *et al.*, 2009) (Figura 1).

Muitos estudos de comportamento em que esse sistema é utilizado, o número de células de cria no favo é empregado como parâmetro para se quantificar o desenvolvimento da colônia (número total de células ou acréscimo de células/tempo). Para isso, os favos são periodicamente analisados e é realizada a contagem manual de células (por meio de desenhos ilustrativos ou fotos). No entanto, esse procedimento básico demanda muito tempo, sendo que uma ferramenta que otimizasse a contagem das células por favo seria de grande utilidade.

Dessa forma, pretendo criar uma função no R que receberia dados sobre o perímetro de um favo (linha de células que forma a borda) e retornasse o número total de células nesse favo. Eterovic *et al.* (2009) utilizaram uma representação do favo a partir de linhas e colunas, em que cada célula é rodeada por outras seis células (como ocorreria com hexágonos) e em que cada célula possui uma coordenada x e y (Figura 2). A partir dessas coordenadas de linhas e colunas, pretendo realizar a entrada de dados (via Excel ou direto no R – a decidir) e a soma de células poderia ser feita a partir da soma parcial por linhas.

Caso isso seja factível, a função poderia ainda conter um argumento para exclusão de buracos (células que foram destruídas) ou ainda realizar uma série temporal da construção de células para uma mesma colônia.



Figs. 1 e 2 (2: modificado de Eterovic *et al.*, 2009)

Plano B

O plano B ainda precisa ser melhor elaborado, mas a ideia principal consiste em analisar possíveis fatores relacionados à produção de sexuais (machos e rainhas) em abelhas sem ferrão, tema do meu mestrado. Estou realizando um experimento em que 3 tratamentos em relação a quantidade de recurso alimentar são oferecidos às colônias: 2X, X e 0 (controle) e verifico a produção dos sexuais para cada colônia. A hipótese é que o alimento influencie a produção de sexuais de forma direta, ou seja, mais recursos resultarão em mais sexuais. No entanto, outro fator que pode ter influência é o tamanho de cada colônia (número de indivíduos, o que pode ser medido pelo número de células construídas). Nesse caso, espera-se que uma colônia com maior número de indivíduos teria maior produção de sexuais. Assim, gostaria de criar uma função que recebesse os dados de número de células de cria (fator por sua vez também dependente do alimento) e a partir de um modelo (baseado em dados de literatura, provavelmente) retornasse uma quantidade de sexuais correspondente ao dado de entrada.

Comentários das propostas (Leo)

A proposta A parece ser útil e pode ser aplicada por outros pesquisadores que tem o mesmo problema. Existem funções no R que criam grid hexagonais. Uma alternativa possível, para resolver teu problema de forma elegante, seria trabalhar com estes grids hexagonais, chamando o pacote e a função dentro da sua função. Acho que vale procurar estes pacotes e ver se ele terá utilidade pra você.

Função units

Fiz uma função que cumpre o objetivo do Plano A. Porém, não consegui fazer isso entrando as coordenadas de cada célula, como pretendia. Tentei a sugestão do Leo, de plotar grids de hexágonos (seria ótimo!), mas não consegui fazer isso de forma compatível com os dados utilizados, pois precisaria do número de hexágonos como entrada (tentei vários pacotes: grid, hexbin, sp...). A função final, units, foi criada de forma mais abrangente, e além de quantificar número de células em um favo, pode calcular a área de objetos de uma imagem, desde que haja uma escala de referência.

Arquivos necessários para os exemplos do help: [Exemplos do Help](#)

Página de Ajuda

units

package:unknown

R Documentation

Contagem de unidades de área em uma imagem

Description:

A função carrega uma imagem no formato jpeg no dispositivo gráfico do R e retorna o número de unidades de área (unidades definidas pelo usuário) de um objeto.

Usage:

```
units(image, n=25, round=TRUE)
```

Arguments:

`image` nome do arquivo da imagem em formato jpeg (ex: "image.jpg")
`n` número de unidades que formam uma área padrão conhecida (utilizada para definição de uma unidade de área, ver "Detalhes").
default: `n=25` (nesse caso, marcar área correspondente a 25 unidades)
`round` se FALSE, não arredonda resultado final

Details:

Função interativa (em dois passos) para cálculo de área.

- 1) A partir da imagem, o usuário deve marcar o perímetro do objeto a ser analisado, por cliques no dispositivo gráfico (o polígono formado não precisa ser fechado, a função faz isso).
- 2) Cálculo da unidade de área para referência, feito a partir de uma área de amostra com número de unidades de área conhecido (novo polígono marcado pelo usuário, deve conter "n" unidades, definido no argumento "n").

Value:

Número de unidades que formam o objeto marcado na imagem.

Warning:

A marcação de pontos no dispositivo gráfico é feita com cliques do mouse. A finalização da interação depende da interface utilizada. No console do R, finalizar com botão direito do mouse. No RStudio, utilizar a tecla "esc" ou com o botão "finish" do dispositivo.

Caso o perímetro contenha um grande número de pontos, o processamento pode levar certo tempo.

Note:

A função foi criada inicialmente para contar a quantidade de células de cria em um favo de abelha. No entanto, pode ser utilizada para calcular a área de objetos diversos, desde que a imagem também contenha uma escala de referência.

Pacotes requeridos para funcionamento: `ReadImages` e `splancs`.

Author(s):

Sheina Koffler
sheinak@gmail.com

Função `getpoly{splancs}` - modificada: (Original por Barry Rowlingson, Peter Diggle, Roger Bivand, Giovanni Petris e Stephen Eglen).

References:

Pacote `ReadImages`:

<http://cran.r-project.org/web/packages/ReadImages/index.html>

Autor: Markus Loecher

Pacote `splancs`: <http://cran.r-project.org/web/packages/splancs/index.html>

Autores: Barry Rowlingson, Peter Diggle, Roger Bivand, Giovanni Petris e Stephen Eglén.

Examples:

Para rodar os exemplos, é necessário baixar os arquivos das imagens (disponível em: http://ecologia.ib.usp.br/bie5782/doku.php?id=bie5782:01_curso_atual:alunos:trabalho_final:sheina.koffler:start)

```
####contagem de células em favo de abelha
units("favo1.jpg", n=49)
#1ª interação: pontos marcando perímetro do favo (apenas células fechadas)
#2ª interação (amostra de área): marcar polígono contendo 49 unidades
(losango de 7 x 7 células)
#retorna número de células que o favo contém
#o valor real da contagem é 211 células

####cálculo da área de uma caixa
units("dove.jpg", n=1, round=FALSE)
#1ª interação: pontos marcando perímetro da caixa
#2ª interação (amostra de área): marcar polígono contendo 1 cm2 (atenção:
são 4 quadradinhos da referência de papel quadriculado)
#retorna valor da área aproximada da caixa, em cm2 (unidade de referência)
#a área real é 42.3 cm2
```

Código da Função

```
units <- function(image, n=1, round=TRUE)
{
  library("ReadImages")
  library("splancs")
  obj <- read.jpeg(image)
  obj
  plot(obj)
  ##funcao "getpoly"(splancs) com avisos modificados e traduzidos, objetos
modificados
{
  cat("\t Função interativa! \n")
  cat("\t Com o botão esquerdo do mouse, plote os pontos da borda do
objeto. \n")
  cat("\t Não feche o polígono formado, a função faz isso para você. \n")
  cat("\t Para finalizar, clique com o botão direito do mouse. \n")
  cat("\t Tenha calma... pode demorar um pouco para acabar. \n\n")
}
  pontos <- locator(type = "l")
  poly <- cbind(pontos$x, pontos$y)
  lines(c(pontos$x[1], pontos$x[length(pontos$x)]), c(pontos$y[1],
pontos$y[length(pontos$y)]))
  poly
```

```
area.total <- areapl(poly)
##funcao "getpoly"(splancs) com avisos modificados e traduzidos, objetos
modificados
{
  cat("\t Função interativa! (de novo) \n")
  cat("\t Criação de um polígono para amostra. \n")
  cat("\t Com o botão esquerdo do mouse, plote um polígono de área
conhecida (",n, "unidade(s)). \n")
  cat("\t Não feche o polígono, a função faz isso para você. \n")
  cat("\t Para finalizar, clique com o botão direito do mouse.\n\n")
}
pontos.2 <- locator(type = "l")
poly.sample <- cbind(pontos.2$x, pontos.2$y)
lines(c(pontos.2$x[1], pontos.2$x[length(pontos.2$x)]), c(pontos.2$y[1],
pontos.2$y[length(pontos.2$y)]))
poly.sample
area.sample <- areapl(poly.sample)
area.unit <- area.sample/n
units <- area.total/area.unit
if(round==TRUE)
{result <- round(units)}
if(round==FALSE)
{result <- units}
return(result)
}
```

Arquivo da Função

[units.r](#)

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2013:alunos:trabalho_final:sheina.koffler:start



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