

TUTORIAL DE ANÁLISES PRELIMINARES E GRÁFICAS NO R PARA MODELOS DE NICHOS ECOLÓGICOS



unesp



Maurício Humberto Vancine – mauricio.vancine@gmail.com

Ecólogo – Mestrando em Ciências Biológicas (Zoologia)

Laboratório de Ecologia Espacial e Conservação (LEEC)

Laboratório de Herpetologia

UNESP – Rio Claro



ORCID QR

Rio Claro - 2016

Prefácio

Esse tutorial foi escrito em dezembro de 2016, durante a disciplina “**Modelagem de Distribuição de Espécies para a Conservação da Biodiversidade**” (UNESP – Rio Claro), dos **Programas de Pós-Graduação em Ciências Biológicas (Zoologia), UNESP – Câmpus de Rio Claro e Programa de Pós-Graduação em Recursos Florestais (PPGRF), LCF/ESALQ/USP – Piracicaba**, lecionada pelos professores Dra. Katia M. P. M. de Barros Ferraz (ESALQ/USP), Dr. Milton Cezar Ribeiro (UNESP - Rio Claro) e Dr. Thadeu Sobral-Souza (UNESP - Rio Claro).

O objetivo desse tutorial é fornecer às pessoas que estão iniciando nessa área de modelagem, uma base para começar a utilizar o Software R no processo de modelagem de nicho ecológico (Ecological Niche Modeling – ENM). Esse tutorial aborda os seguintes tópicos: (1) estrutura e manejo de arquivos raster e vetor no R; (2) análises preliminares das variáveis ambientais: Correlação de Pearson, PCA e Análise Fatorial; (3) análises gráficas posteriores dos resultados dos ENMs.

Todos os passos desse tutorial foram realizados no Software R, versão 3.3.1 (www.r-project.org, R Core Team 2016), para o sistema Windows, de modo que pode haver alguma divergência para os sistemas Linux e Mac. Pressupõe-se que o usuário desse tutorial possua certa familiaridade com o software R. Para realizar os exemplos, foi utilizado dados de variáveis bioclimáticas (mais informações podem ser obtidas pelo site: <http://www.worldclim.org/bioclim> e também pelo trabalho de Hijmans et al. 2005) de temperatura e precipitação, de 1950-1999, para a extensão do Neotrópico, com resolução espacial de 50 km, para facilitar os exemplos, disponíveis no site: <http://ecoclimate.org> (Lima et al. 2015, Lima et al. 2016). Além disso, foi construído um ENM no software MaxEnt, versão 3.3.3k (<https://www.cs.princeton.edu/~schapire/maxent/>, Phillips, Dudík & Schapire 2004), com dados de ocorrência de *Bromelia balansae*.

A teoria referente aos ENMs pode ser encontrada no livro “Ecological Niches and Geographic Distributions” (Peterson et al. 2011). Todos os códigos do R estão escritos com letras diferenciadas, no caso Courier New. Por fim, esse tutorial pode e deve ser lido rodando os comandos de três scripts (texto com os códigos), um para cada tópico que abordado e um conjunto de dados para todos eles.

Qualquer dúvida ou erro que eu possa ter cometido, não se acanhe, me mande um e-mail (mauricio.vancine@gmail.com), responderei assim que possível =]

Sumário

- 1 Estrutura e manejo de arquivos raster e vetor no R
 - 1.1 Pacotes necessários
 - 1.2 Raster
 - 1.3 Vetor
 - 1.4 Extrair valores do raster
 - 1.5 Extrair valores do raster para arquivos vetor
 - 1.6 Definir a extensão (máscara) do raster a partir de um arquivo vetor

- 2 Análises preliminares das variáveis ambientais
 - 2.1 Pacotes necessários
 - 2.2 Preparação dos dados
 - 2.3 Correlação de Pearson
 - 2.4 Análise de Componentes Principais (PCA)
 - 2.5 Análise Fatorial

- 3 Análises gráficas posteriores dos resultados dos ENMs
 - 3.1 Pacotes necessários
 - 3.2 Plot da adequabilidade de saída do MaxEnt e variáveis ambientais
 - 3.3 Corte pelo limite de corte (threshold)
 - 3.4 Histograma
 - 3.5 Gráfico de dispersão (scatterplot)

Referências

1 Estrutura e manejo de arquivos raster e vetor no R

Início esse tutorial abordando os principais tipos de arquivos SIG utilizado nos ENMs (raster e vetor). Geralmente os arquivos são grandes e levam bastante tempo para ser importados para o R. Por isso, certifique-se que o computador que irá rodar os scripts possui memória RAM e processador suficientes para realizar esse processamento.

Há diversos materiais disponíveis para se trabalhar com arquivos nos formatos raster e vetor no R, no entanto, acredito que três deles sejam primordiais no início do aprendizado.

1 Landscape Ecology in R - Dr. Felipe Martello (UNESP - Rio Claro)
http://www.leec.eco.br/pdfs/Support_Landscape_ecology_R_2016_06_15.pdf%20

2 Spatial data manipulation - Prof. Robert Hijmans
<http://rspatial.org/spatial/index.html>

3 Species distribution modeling with R - Prof. Robert J. Hijmans e Profa. Jane Elith
<https://cran.r-project.org/web/packages/dismo/vignettes/sdm.pdf>

Para entender melhor os formatos dos arquivos e como se trabalhar com 'Sistema de Informação Geográfica' (SIG) em softwares como ArcGIS e QGIS, recomendo a leitura dessa apostila de SIG.

4 Geoprocessamento aplicado a Biologia - Dr. Ubirajara de Oliveira (UFMG)
<https://www.dropbox.com/s/7vlxktpga5rrsfu/Geoprocessamento%20aplicado%20a%20Biologia.pdf?dl=0>

1.1 Pacotes necessários

Para a realização dessa parte do tutorial, alguns pacotes devem ser instalados e carregados no software R. Os pacotes utilizados serão: “raster” e “rgdal”, que permitem a operação de arquivos raster e vetor no R.

```
### 1.1 pacotes necessarios
# instalar os pacotes necessarios
install.packages(c("raster", "rgdal"), dep = T)

# carregar os pacotes necessarios
library(raster)
library(rgdal)
```

1.2 Raster

Nesta seção, vamos importar um ou mais arquivos raster para o R, utilizando os exemplos com as duas extensões mais comuns (.asc e .tif) e explorar algumas propriedades.

Para importar um ou mais arquivos raster, utilizares a função 'raster', do pacote “raster”.

```
### 1.2 raster
# importar um arquivo raster
# asc
# diretorio da pasta com os arquivos .asc
setwd("")

# importar um arquivo .asc no formato rasterlayer
bio1.asc <- raster("bio01_neotropic_50km_gcs_wgs84.asc")

# visualizar as propriedades do arquivo raster importado
bio1.asc

class      : RasterLayer
dimensions : 168, 154, 25872  (nrow, ncol, ncell)
resolution : 0.5, 0.5  (x, y)
extent     : -112, -35, -55.5, 28.5  (xmin, xmax, ymin, ymax)
coord. ref.: NA
data source :
names      : bio01_neotropic_50km_gcs_wgs84

# plot do raster
plot(bio1.asc)
plot(bio1.asc, col = rainbow(n = 50, alpha = .5))
plot(bio1.asc, col = heat.colors(50, .8))
plot(bio1.asc, col = terrain.colors(50, .8))
plot(bio1.asc, col = topo.colors(50, .8))
plot(bio1.asc, col = cm.colors(50))
```

As funções 'rainbow', 'heat.colors', 'terrain.colors', 'topo.colors' e 'cm.colors' geram cores diversas para os pixels no argumento 'col' da função 'plot'. Além disso, essas funções possuem basicamente dois argumentos: 1. 'n', que informará o número de cores para o plot e 2. 'alpha', que informará, numa escala de 0 a 1, a transparência da cor.

Para importar um arquivo .tif, basta mudar o formato no script.

```
# tif
# diretorio da pasta com os arquivos .tif
setwd("")

# importar um arquivo .tif no formato rasterlayer
bio1.tif <- raster("bio01_neotropic_50km_gcs_wgs84.tif")

# visualizar as propriedades do arquivo raster importado
bio1.tif

# plot do raster
plot(bio1.tif)

# visualizar as propriedades do arquivo raster importado
bio1.tif
class      : RasterLayer
dimensions : 168, 154, 25872  (nrow, ncol, ncell)
resolution : 0.5, 0.5  (x, y)
extent     : -112, -35, -55.5, 28.5  (xmin, xmax, ymin, ymax)
coord. ref.: +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0
data source :
names      : bio01_neotropic_50km_gcs_wgs84
```

Atentar para as diferenças das características dos dois tipos de formato de arquivo raster importados: o arquivo tipo .asc não possui projeção definida, apesar de não haver problemas de compatibilidade de projeção para outras operações. O arquivo .tif possui projeção definida, como podemos notar em `coord. ref.`

Para acessar a projeção dos arquivos, pode-se utilizar as funções 'crs' ou 'projection'.

```
# projecao dos arquivos
crs(bio1.asc)
crs(bio1.tif)
projection(bio1.tif)
```

Na maior parte das vezes temos de importar diversos arquivos raster para o R. Podemos utilizar o formato anterior e importar diversos arquivos separadamente, atribuindo cada um a uma variável diferente ou podemos utilizar a função 'stack' do pacote “raster”, como veremos adiante.

Para que essa parte funcione corretamente, é necessário que todos os arquivos raster estejam na mesma pasta informada para o R na função 'setwd'.

```
# asc
# diretorio da pasta com os arquivos .asc
setwd("")

# listar os nomes dos arquivos na pasta do diretorio
list.files(pattern = ".asc")
asc <- list.files(pattern = ".asc")

# importar os arquivos .asc no formato rasterstack
asc.bios <- stack(asc)
asc.bios

# renomear os arquivos raster importados
names(asc.bios) <- paste0("bio", 1:19)
asc.bios

# plot de todos dos raster
plot(asc.bios)

# plot de alguns dos raster
plot(asc.bios[[1]])
plot(asc.bios[[1:10]])
plot(asc.bios[[c(2, 5, 17, 18)])])
```

Para importar os arquivos raster com a extensão .tif, basta mudar a extensão no script.

```
# tif
# diretorio da pasta com os arquivos .tif
setwd("")
```

```
# listar os nomes dos arquivos na pasta do diretorio
list.files(pattern = ".tif")
tif <- list.files(pattern = ".tif")

# importar os arquivos .tif no formato rasterstack
tif.bios <- stack(tif)
tif.bios

# renomear os arquivos raster importados
names(tif.bios) <- paste0("bio", 1:19)
tif.bios

# plot de todos dos raster
plot(tif.bios)

# plot de alguns dos raster
plot(tif.bios[[c(2, 5, 17, 18)]], col = rainbow(100, .7))
```

1.3 Vetor

Nessa seção vamos importar três tipos de arquivos vetor para o R e explorar algumas de suas propriedades.

Utilizaremos a função 'shapefile' do pacote “raster”.

```
### 1.3 vetor
# importar arquivos vetor
# diretorio da pasta com os arquivos vetor
setwd("")

# pontos
pto <- shapefile("bromelia_balansae.shp")

# plot
plot(pto)
plot(pto, pch = 16)
plot(pto, pch = 16, cex = 1.5)
plot(pto, pch = 16, cex = 1.5, axes = T)
```


Tutorial de Análises Preliminares e Gráficas no R para Modelos de Nicho Ecológico

```
# linhas
lin <- shapefile("rios_neotropico_gcs_wgs84.shp")

# plot
plot(lin, col = "blue", axes = T)

# adicionar plot de pontos ao plot de linhas
plot(pto, cex = 1.5, add = T)
points(pto, pch = 16, cex = .8, col = "red")

# poligonos
pol <- shapefile("países_neotropico_gcs_wgs84.shp")

# plot
plot(pol, axes = T)
plot(pol, col = "gray90", axes = T)

# adicionar plot de pontos e linhas
plot(lin, col = "blue", add = T)
plot(pto, cex = 1.5, pch = 20, col = "red", add = T)

# plot raster e adicionar pontos, linhas e poligonos
plot(tif.bios[[1]])
plot(pto, cex = 1.5, pch = 20, add = T)
plot(lin, col = "blue", add = T)
plot(pol, lwd = .01, add = T)
```

Os arquivos vetor possuem associado a eles uma tabela de atributos, semelhante a uma planilha eletrônica tipo Calc ou Excel, onde cada linha da tabela é associada a uma feição, como um ponto, linha ou polígono. Podemos acessar essa tabela e escolher uma ou mais feições de interesse.

```
# acessar tabela de atributos
pto@data
head(pto@data)
dim(pto@data)

lin@data
dim(lin@data)
```

```
pol@data
head(pol@data)
dim(pol@data)

# acessar linhas e escolher por linhas
pto1 <- pto[1, ] # primeira linha
plot(pto1, axes = T)

pto2 <- pto[1:10, ] # linha 1 a linha 10
plot(pto2, axes = T)

pto3 <- pto[c(1, 5, 10), ] # linhas 1, 5 e 10
plot(pto3, axes = T)

pto4 <- pto[pto$Lat > -22, ] # linhas com latitude acima de -22
plot(pto4, axes = T)

pto5 <- pto[sample(1:40, 20), ] # 20 linhas aleatorias
plot(pto5, axes = T)

lin1 <- lin[lin$SYSTEM == 'Parana', ] # seleciona a linha com o SYSTEM igual a Parana
plot(lin1, axes = T, col = "blue")

poll <- pol[pol$COUNTRY == 'Brazil', ] # seleciona a linha com o COUNTRY igual Brazil
plot(poll, axes = T)
```

1.4 Extrair valores do raster

Para extrair valores de todos os pixels de um ou mais arquivos raster .asc ou .tif, iremos utilizar a função 'values', do pacote "raster", sendo que as operações são semelhantes para os dois formatos e usarei apenas o arquivo .asc como exemplo.

```
### 1.4 extrair valores do raster
# apenas um arquivo raster
# variavel importada
bio1.asc
```

```
# plot
plot(bio1.asc)

# extrair valores
bio1.v <- values(bio1.asc)
bio1.v # possui ainda NAs
length(bio1.v) # comprimento do vetor com os NAs

# retirar os NAs
bio1.v.na <- na.omit(bio1.v)
bio1.v.na
length(bio1.v.na) # comprimento do vetor sem os NAs

# varios arquivos raster
# variavel importada
asc.bios

# plot
plot(asc.bios)

# extrair valores
asc.bios.v <- values(asc.bios)
head(asc.bios.v) # possui ainda NAs
dim(asc.bios.v) # dimensao da matriz com os NAs

# retirar os NAs
asc.bios.v.na <- na.omit(asc.bios.v)
head(asc.bios.v.na)
dim(asc.bios.v.na) # dimensao da matriz sem os NAs
```

1.5 Extrair valores do raster para os arquivos vetor

Algumas operações demandam a utilização dos valores dos pixels apenas para os arquivos vetor. Para realizar essa operação, utilizaremos a função 'extract' do pacote “raster”, onde o primeiro argumento é o arquivo raster e o segundo é o arquivo vetor.

Tutorial de Análises Preliminares e Gráficas no R para Modelos de Nicho Ecológico

```
### 1.5 extrair valores do raster para os arquivos vetor
# apenas um arquivo raster
# variavel importada
biol.asc

# plot
plot(biol.asc)
plot(pto, pch = 20, add = T)

# extrair valores para o vetor de pontos
biol.pto <- extract(biol.asc, pto)
biol.pto

# plot
plot(biol.asc)
plot(lin, col = "blue", add = T)

# extrair valores para o vetor de linhas
biol.lin <- extract(biol.asc, lin)
biol.lin # saida e uma lista, onde cada elemento da lista possui os valores do raster
para cada feicao da linha e a ordem da lista segue a mesma ordem da tabela de
atributos

# plot
plot(biol.asc)
plot(pol, add = T)

# extrair valores para o vetor de poligonos
biol.pol <- extract(biol.asc, pol) # esse comando ira demorar muito tempo!
mean(biol.pol) # saida e uma lista, onde cada elemento possui os valores para cada
poligono e a ordem segue a tabela de atributos

# varios arquivos raster
# variaveis importadas
asc.bios
```

```
# plot
plot(asc.bios)

# extrair valores para o vetor de pontos
asc.bios.pto <- extract(asc.bios, pto)
head(asc.bios.pto) # matriz com os valores de todas as variaveis

# extrair valores para o vetor de linhas
asc.bios.lin <- extract(asc.bios, lin) # esse comando ira demorar muito tempo!
head(asc.bios.lin[[1]]) # saida e uma lista, onde cada elemento possui os valores
para cada linha de todas as variaveis na forma de matriz

# extrair valores para o vetor de poligonos
asc.bios.pol <- extract(asc.bios, pol) # nao rode esse comando, ira demorar muito tempo!
head(asc.bios.pol[[1]]) # saida e uma lista, onde cada elemento possui os valores
para cada poligono de todas as variaveis na forma de matriz
```

1.6 Definir a extensão (máscara) do raster a partir de arquivo vetor

Para construir os ENMs deve-se atentar para a extensão dos arquivos raster utilizados, de modo que há artigos discutindo toda essa problemática de se utilizar extensões com diferentes tamanhos e seus efeitos (e.g. Barve et al. 2011).

Além disso, quando baixamos as variáveis climáticas, geralmente bioclimáticas (Hijmans et al. 2005), elas estão disponíveis para o mundo todo e devemos ajustar a extensão dessas variáveis à nossa pergunta.

Para realizar essa operação será utilizado duas funções do pacote “raster” : 'crop' e 'mask', onde o primeiro argumento é o arquivo raster e o segundo é o arquivo vetor, para ambas as funções. Novamente, irei demonstrar as funções para um arquivo e depois para diversos arquivos, apenas para arquivos no formato raster .asc. Mas antes, devemos importar o arquivo vetor para limitar o arquivo raster. Por fim, iremos exportar e criar novas variáveis raster a partir desse ajuste.

```
### 1.6 definir a extensao (mascara) do raster a partir de arquivo vetor
# apenas um arquivo raster
# variavel
biol.asc
```

Tutorial de Análises Preliminares e Gráficas no R para Modelos de Nicho Ecológico

```
# plot
plot(biol.asc)

# importar o shapefile do brasil
# diretorio da pasta dos arquivos vetor
setwd("")

br <- shapefile("brasil_gcs_wgs84.shp")
plot(br, axes = T)

# ajustar a extensao e as dimensoes - crop
biol.br.c <- crop(biol.asc, br)
biol.br.c
plot(biol.br.c, axes = T)
plot(br, add = T)

# ajustar apenas para dentro do arquivo shape - mask
biol.br <- mask(biol.br.c, br)
biol.br
plot(biol.br, axes = T)

# verificar a diferenca
plot(biol.asc, axes = T)
plot(br, axes = T, add = T)

plot(biol.br.c, axes = T)
plot(br, axes = T, add = T)

plot(biol.br, axes = T)
plot(br, axes = T, add = T)

# exportar o arquivo criado
# diretorio da pasta de saida
setwd("")

# exportando no formato .asc e .tif
writeRaster(biol.br, "biol_br.asc", format = "ascii")
writeRaster(biol.br, "biol_br.tif", format = "GTiff") # sem projecao definida
```

```
# varios arquivos raster
# variaveis importadas
asc.bios

# plot
plot(asc.bios)

# ajustar a extensao e as dimensoes - crop
bios.br.c <- crop(asc.bios, br)
bios.br.c
plot(bios.br.c, axes = T)

# ajustar apenas para dentro do arquivo shape - mask
bios.br <- mask(bios.br.c, br)
bios.br
plot(bios.br, axes = T)

# verificar a diferenca
par(mfrow = c(3, 1))
plot(asc.bios[[1]], axes = T)
plot(br, axes = T, add = T)

plot(bios.br.c[[1]], axes = T)
plot(br, axes = T, add = T)

plot(bios.br[[1]], axes = T)
plot(br, axes = T, add = T)

# exportar um arquivo criado
# diretorio da pasta de saida
setwd("")

# exportando um arquivo no formato .asc e .tif
writeRaster(bios.br$bio1, "bio1_br.asc", format = "ascii")
writeRaster(bios.br$bio1, "bio1_br.tif", format = "GTiff") # sem projecao
```

Tutorial de Análises Preliminares e Gráficas no R para Modelos de Nicho Ecológico

```
# exportar todos os arquivos criados
for(i in 1:length(names(bios.br))){
  writeRaster(bios.br[[i]], paste0("bio", i, "_br.asc"), format = "ascii")}

for(i in 1:length(names(bios.br))){
  writeRaster(bios.br[[i]], paste0("bio", i, "_br.tif"), format = "GTiff")} # sem
projecao
```


2 Análises preliminares das variáveis ambientais

Antes de iniciar a construção dos ENMs, é necessário avaliar a colinearidade das variáveis de entrada. Nesse tutorial, exploro quatro alternativas utilizando as variáveis .asc da região Neotropical, especificadas no Prefácio. Para diferentes extensões, as análises de colinearidade têm de ser realizadas novamente, pois os limites das variáveis mudam os valores das variáveis e assim também há mudança da composição do espaço ambiental e por conseguinte da colinearidade das variáveis.

2.1 Pacotes necessários

Os pacotes necessários serão: “raster”, “rgdal”, “corrplot”, “RStoolbox” e “psych”, que permitem a operação de arquivos raster e vetor no R, plotar a correlação e realizar as análises de PCA e Análise Fatorial, respectivamente.

```
### 2.1 pacotes necessarios
# instalar pacotes
install.packages(c("raster", "rgdal", "corrplot", "RStoolbox", "vegan", "psych"),
dep = T)

# carregar pacotes
library("raster")
library("rgdal")
library("corrplot")
library("RStoolbox")
library("vegan")
library("psych")
```

2.2 Preparação dos dados

Nesse tópico apenas retomo os comandos do tópico 1.4.

```
### 2.2 preparacao dos dados
# diretorio
setwd("")

# listar o nome dos arquivos no diretorio com o padrao .asc
asc <- list.files(pattern = ".asc")
asc
```

```
# carregar os arquivos .asc
as <- stack(asc)
as

# renomear as variaveis
names(as)
names(as) <- paste0("bio", 01:19)
names(as)
as

# plot
plot(as)
plot(as[[1]])

# extraindo valores dos arquivos .asc e omitindo os NAs
as.v <- values(as)
as.v.na <- na.omit(as.v)
head(as.v.na)
dim(as.v.na)
```

2.3 Correlação

A correlação é uma das análises mais simples em qualquer análise estatística descritiva. Nesse exemplo, usaremos a correlação de Pearson para as 19 variáveis bioclimáticas. O valor do coeficiente de correlação de Pearson (ρ ou r) mede a relação entre duas variáveis e varia de -1 (negativamente correlacionadas), 0 (sem correlação) e 1 (positivamente correlacionadas). O resultado da análise será uma matriz “triangular” (valores iguais acima e abaixo da diagonal da matriz), relacionando nas linhas e nas colunas as 19 variáveis bioclimáticas.

Geralmente assume-se correlação entre duas variáveis, se as mesmas possuem ρ ou r acima de 0.7 ou abaixo de -0.7. No entanto, isso não é uma regra e outros valores, acima ou abaixo desse limiar podem ser escolhidos, dependendo da pergunta que se tem a responder.

```
### 2.3 correlacao

# criar pasta e definir diretorio para analise exploratoria - correlacao
dir.create("analise_selecao_variaveis") # cria uma pasta no diretorio
setwd("./analise_selecao_variaveis") # muda o diretorio para a pasta criada
```

Tutorial de Análises Preliminares e Gráficas no R para Modelos de Nicho Ecológico

```
dir.create("correlacao") # cria uma pasta no diretorio
setwd("./correlacao") # muda o diretorio para a pasta criada
getwd()

# tabela da correlacao
cor <- cor(as.v.na)
cor
round(cor, 2) # arredondamento dos valores para dois valores decimais

# exportar tabela com a correlacao
write.table(round(cor(as.v), 2), "cor_pres.xls", row.names = T, sep = "\t")
write.table(iffelse(cor(as.v) >= 0.7, "Sim", "Não"), "cor_pres_afirmacao.xls",
row.names = T, sep = "\t")

# plot da correlacao
tiff("cor_ma.tif", width = 18, height = 18, units = "cm", res = 300, compression =
"lzw")
corrplot(cor(as.v.na), type = "lower", diag = F, tl.srt = 45, mar = c(3, 0.5, 2, 1),
title = "Correlações entre variáveis Bioclimáticas")
dev.off()
```

2.4 Análise de Componentes Principais (PCA)

Outra forma reduzir a colinearidade entre as variáveis é através da utilização da Análise de Componentes Principais (PCA). Essa análise tem como objetivo reduzir o número de variáveis, uma vez que as novas variáveis criadas a partir das variáveis originais são ortogonais (relacionam-se a 90°, i.e., são independentes) e ordenar as linhas (observações) de uma matriz a partir das colunas (descritores).

Há duas formas de fazer uso da PCA: 1. selecionar uma variável bioclimática original para cada eixo da PCA, que possuam alta contribuição para cada eixo da PCA; e 2. utilizar os próprios eixos da PCA como novas variáveis independentes, utilizando aquelas que tenham contribuição relativa também alta. Ambas as formas possuem vantagens e desvantagens e cabe ao pesquisador escolher, com base na pergunta que se quer responder com o uso do ENM, a melhor forma de utilizar esse método.

```
# 2.4 pca

# criar pasta e definir diretorio para analise exploratoria - pca
setwd("../") # voltar uma pasta no diretorio
```

Tutorial de Análises Preliminares e Gráficas no R para Modelos de Nicho Ecológico

```
getwd() # conferir o diretorio
dir.create("pca") # criar pasta no diretorio
setwd("./pca") # mudar o diretorio para a pasta criada
getwd() # conferir o diretorio

# pca do pacote "stats"
# pca com normalizacao interna
pca <- prcomp(as.v.na, scale = T)

# contribuicao de cada eixo (eigenvalues - autovalores)
summary(pca)

# grafico de barras com as contribuicoes
screepplot(pca, main = "Contribuição de cada PC", ylab = "Autovalores")
abline(h = 1, col = "red", lty = 2)

tiff("screepplot.tif", width = 18, height = 18, units = "cm", res = 300, compression
= "lzw")
screepplot(pca, main = "Contribuição de cada PC", ylab = "Autovalores")
abline(h = 1, col = "red", lty = 2)
dev.off()

# valores de cada eixo (eigenvectors - autovetores - escores)
pca$x

# relacao das variaveis com cada eixo (loadings - cargas)
pca$rotation[, 1:5]
abs(pca$rotation[, 1:5])

# exportar tabela com a contribuicao
write.table(round(abs(pca$rotation[, 1:5]), 2), "contr_pca.xls", row.names = T, sep
= "\t")

# plot
biplot(pca)
```

```
# pca como novas variaveis
# pca dos raster
pca.as <- rasterPCA(as, spca = T) # esse comando ira demorar
pca.as

# contribuicao dos componentes
summary(pca.as$model)

# grafico de barras com as contribuicoes
screepplot(pca.as$model, main = "Autovalores")
abline(h = 1, col = "red", lty = 2)

tiff("screepplot_raster.tif", width = 20, height = 20, units = "cm", res = 300,
compression = "lzw")
screepplot(pca.as$model, main = "Autovalores")
abline(h = 1, col = "red", lty = 2)
dev.off()

# plot das pcs como novas variaveis
plot(pca.as$map[[1:5]])

# exportar as novas variaveis
setwd("")
for(i in 1:5){
  writeRaster(pca.as$map[[i]], paste0("pc", i, "_br.asc"), format = "ascii")}
```

2.5 Análise Fatorial

A análise fatorial é muito parecida com a PCA, sendo que o objetivo também é selecionar as variáveis pela contribuição de cada variável a cada eixo da fatorial. No entanto, a diferença é que podemos escolher o número de eixos que queremos que as variáveis tenham relação com cada eixo da fatorial, aumentando assim essa relação.

O resultado é uma matriz que relaciona cada variável a cada eixo, sendo que os valores, com esse tipo de variável ambiental, irão variar de -1 a 1. Da mesma forma que na correlação de Pearson, escolhemos os maiores valores absolutos, um para cada eixo, levando em consideração a biologia da espécie. Essa análise, com esse propósito, foi proposta no trabalho do Prof. Thadeu (Sobral-Souza, Lima-Ribeiro & Solferini 2015).

Tutorial de Análises Preliminares e Gráficas no R para Modelos de Nicho Ecológico

```
# 2.5 analise fatorial

# criar pasta e definir diretorio para analise exploratoria - fatorial
setwd("../")
getwd()
dir.create("fatorial")
setwd("./fatorial")
getwd()

# padronizacao dos dados
as.v.na.p <- decostand(as.v.na, method = "standardize")
summary(as.v.na.p)

# analises preliminares de possibilidade de uso da analise fatorial
# kmo e bartlett
KMO(cor(as.v.na.p)) # deve ser acima de 0.5
cortest.bartlett(cor(as.v.na.p), n = nrow(as.v.na)) # deve ser sognificativo (p <
0.05)

# numero de eixos - semelhante a pca
# screeplot
fa <- fa.parallel(as.v.na.p, fm = "ml", fa = "fa") # sugere 5 eixos

# exportar screeplot
tiff("screeplot_fatorial.tif", height = 664, width = 665, unit = "px", res = 96)
fa.parallel(as.v.na.p, fm = "ml", fa = "fa")
dev.off()

# fatorial
fa.as <- fa(as.v.na.p, nfactors = 5, rotate = "varimax", fm = "ml")
as.loadings <- loadings(fa.as)
as.loadings

# exportar tabela dos resultados
write.table(abs(round(as.loadings, 2)), "as_loadings.xls", row.names = T, sep = "\t")

# bios escolhidas
# bio02, bio04, bio10, bio16, bio17
```

Tutorial de Análises Preliminares e Gráficas no R para Modelos de Nicho Ecológico

```
# significado das bios
# BIO1 = Temperatura media anual
# BIO2 = Variacao da media diurna (media por mes (temp max - temp min))
# BIO3 = Isotermalidade (BIO2/BIO7) (* 100)
# BIO4 = Sazonalidade da temperatura (desvio padrao deviation *100)
# BIO5 = Temperatura maxima do mes mais quente
# BIO6 = Temperatura minima do mes mais frio
# BIO7 = Variacao da temperatura anual (BIO5-BIO6)
# BIO8 = Temperatura media do trimestre mais chuvoso
# BIO9 = Temperatura media do trimestre mais seco
# BIO10 = Temperatura media do trimestre mais quente
# BIO11 = Temperatura media do trimestre mais frio
# BIO12 = Precipitacao anual
# BIO13 = Precipitacao do mes mais chuvoso
# BIO14 = Precipitacao do mes mais seco
# BIO15 = Sazonalidade da precipitacao (coeficiente de variacao)
# BIO16 = Precipitacao do trimestre mais chuvoso
# BIO17 = Precipitacao do trimestre mais seco
# BIO18 = Precipitacao do trimestre mais quente
# BIO19 = Precipitacao do trimestre mais frio

###-----###
```

3 Gráficos posteriores com os resultados dos ENMs

Para a realização dessa última parte do tutorial, usarei os resultados de um ENM que construí no Software MaxEnt (versão 3.3.3k (<https://www.cs.princeton.edu/~schapire/maxent/>), Phillips, Dudík & Schapire 2004), com dados de ocorrência de *Bromelia balansae* e as variáveis Bio02, Bio04, Bio10, Bio16 e Bio17, resultado da Análise Fatorial (tópico 2.4).

Utilizei os seguintes parâmetros do software: 1. partição de 70% do conjunto de dados para treino e 30% para teste, 2. método de amostragem por 'Bootstrap', 3. 10 réplicas aleatórias com substituição, e todos os outros parâmetros deixei no 'default'.

Utilizaremos para as análises, os arquivos de saída .dat dos gráficos das variáveis utilizadas para construir o modelo, o arquivo .asc de saída da média dos 10 modelos, o arquivo de saída do desvio padrão entre os 10 modelos, dois arquivos .asc (Bio01 e Bio12) de variáveis que não entraram na construção do modelo e o arquivo .shp de pontos de ocorrência.

3.1 Pacotes necessários

Os pacotes utilizados serão: 'raster' e 'rgdal', que permitem a operação de arquivos raster e vetor no R e o pacote 'ggplot2', que permite criar gráficos diferenciados.

```
### 3.1 pacotes necessarios
# instalar pacotes
install.packages(c("raster", "rgdal", "ggplot2"), dep = T)

# carregar pacotes
library("raster")
library("rgdal")
library("ggplot2")
```

3.2 Plot da adequabilidade de saída do MaxEnt e variáveis ambientais

Esses gráficos são gerados na saída da pasta do MaxEnt. No entanto, podemos utilizar o arquivo .dat para criar e editar esses gráficos. Utilizaremos os arquivos que possuam final '_only.dat'.

```
### 3.2 plot da adequabilidade e variaveis de saida do maxent
# diretorio dos plots da saida do maxent
setwd("")
```


Tutorial de Análises Preliminares e Gráficas no R para Modelos de Nicho Ecológico

```
# listar os arquivos
dat <- list.files(pattern = ".dat")
dat

# carregar os dados
adeq <- list()

for(i in dat){
  adeq[[i]] <- read.table(i, h = T, sep = ",", dec = ".")
}

head(adeq[[1]])
head(adeq[[2]])
head(adeq[[3]])
head(adeq[[4]])
head(adeq[[5]])

# diretorio de saida
setwd("../")
dir.create("graficos_saidas")
setwd("./graficos_saidas")
getwd()

# graficos
tiff("bio2.tif", width = 18, height = 18, units = "cm", res = 300, compression =
"lzw")
plot(y ~ x,
      data = adeq[[1]], # dados
      type = "l", # plota a linha
      lwd = 5, # largura da linha
      ylim = c(0, 1), # limite do y
      col = "red", # cor da linha
      ylab = "", # nome do y
      xlab = "Bio 2 - Variação da média diurna", # nome do x
      main = "", # titulo
      col.axis = "grey30", # cor do eixo
      cex.lab = 1.5, # tamanho da fonte do nome dos eixos
      cex.main = 1.5, # tamanho da fonte do nome do titulo)
```

Tutorial de Análises Preliminares e Gráficas no R para Modelos de Nicho Ecológico

```
cex.axis = 1.15, # tamanho dos numeros do eixo
bty = "l", # tipo caixa em volta
las = 1) # inverte a fonte do eixo y
mtext("Adequabilidade", side = 2, line = 2.6, cex = 1.5)
dev.off()
```

```
tiff("bio4.tif", width = 18, height = 18, units = "cm", res = 300, compression =
"lzw")
```

```
plot(y ~ x,
      data = adeq[[2]], # dados
      type = "l", # plota a linha
      lwd = 5, # largura da linha
      ylim = c(0, 1), # limite do y
      col = "red", # cor da linha
      ylab = "", # nome do y
      xlab = "Bio 4 - Sazonalidade da temperatura", # nome do x
      main = "", # titulo
      col.axis = "grey30", # cor do eixo
      cex.lab = 1.4, # tamanho da fonte do nome dos eixos
      cex.main = 1.5, # tamanho da fonte do nome do titulo
      cex.axis = 1.15, # tamanho dos numeros do eixo
      bty = "l", # tipo caixa em volta
      las = 1) # inverte a fonte do eixo y
mtext("Adequabilidade", side = 2, line = 2.6, cex = 1.5)
dev.off()
```

```
tiff("bio10.tif", width = 18, height = 18, units = "cm", res = 300, compression =
"lzw")
```

```
plot(y ~ x,
      data = adeq[[3]], # dados
      type = "l", # plota a linha
      lwd = 5, # largura da linha
      ylim = c(0, 1), # limite do y
      col = "red", # cor da linha
      ylab = "", # nome do y
      xlab = "Bio 10 - Temperatura média do trimestre mais quente", # nome do x
      main = "", # titulo
      col.axis = "grey30", # cor do eixo
```

Tutorial de Análises Preliminares e Gráficas no R para Modelos de Nicho Ecológico

```
cex.lab = 1.4, # tamanho da fonte do nome dos eixos
cex.main = 1.5, # tamanho da fonte do nome do titulo
cex.axis = 1.15, # tamanho dos numeros do eixo
bty = "l", # tipo caixa em volta
las = 1) # inverte a fonte do eixo y
mtext("Adequabilidade", side = 2, line = 2.6, cex = 1.5)
dev.off()
```

```
tiff("bio16.tif", width = 18, height = 18, units = "cm", res = 300, compression =
"lzw")
```

```
plot(y ~ x,
      data = adeq[[4]], # dados
      type = "l", # plota a linha
      lwd = 5, # largura da linha
      ylim = c(0, 1), # limite do y
      col = "red", # cor da linha
      ylab = "", # nome do y
      xlab = "Bio 16 - Precipitacao do trimestre mais chuvoso", # nome do x
      main = "", # titulo
      col.axis = "grey30", # cor do eixo
      cex.lab = 1.4, # tamanho da fonte do nome dos eixos
      cex.main = 1.5, # tamanho da fonte do nome do titulo
      cex.axis = 1.15, # tamanho dos numeros do eixo
      bty = "l", # tipo caixa em volta
      las = 1) # inverte a fonte do eixo y
mtext("Adequabilidade", side = 2, line = 2.6, cex = 1.5)
dev.off()
```

```
tiff("bio17", width = 18, height = 18, units = "cm", res = 300, compression = "lzw")
```

```
plot(y ~ x,
      data = adeq[[5]], # dados
      type = "l", # plota a linha
      lwd = 5, # largura da linha
      ylim = c(0, 1), # limite do y
      col = "red", # cor da linha
      ylab = "", # nome do y
      xlab = "'Bio 17 - Precipitacao do trimestre mais seco", # nome do x
      main = "", # titulo
```

```
col.axis = "grey30",           # cor do eixo
cex.lab = 1.4,                 # tamanho da fonte do nome dos eixos
cex.main = 1.5,                # tamanho da fonte do nome do titulo
cex.axis = 1.15,               # tamanho dos numeros do eixo
bty = "l",                     # tipo caixa em volta
las = 1)                       # inverte a fonte do eixo y
mtext("Adequabilidade", side = 2, line = 2.6, cex = 1.5)
dev.off()
```

3.3 Corte pelo limite de corte (*threshold*)

Depois que os modelos são gerados, deve-se fazer um “corte” por alguns limites de corte (*thresholds*), baseados em diversas estatísticas. Não entrarei em detalhes, sendo que apresentarei apenas os quatro limites mais utilizados: 1. Minimum training presence logistic threshold, 2. 10 percentile training presence logistic threshold, 3. Maximum test sensitivity plus specificity logistic threshold e 4. Lowest presence threshold (Pearson et al. 2007). Mais detalhes podem ser vistos no livro “Ecological Niches and Geographic Distributions” (Peterson et al. 2011).

```
### 3.3 corte pelo threshold

# diretorio com a saida do maxent
setwd("")

# importar os dados
# listas os arquivos na pasta
enm <- raster("B.balansae_avg.asc")
enm

# plot
plot(enm)

# exportar e criar o arquivo como .tif
writeRaster(enm, "enm.tif", format = "GTiff")

# importar .tif criado
enm.tif <- raster("enm.tif")
enm.tif
```

Tutorial de Análises Preliminares e Gráficas no R para Modelos de Nicho Ecológico

```
# definir projecao
projection(enm.tif) <- "+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0"
enm.tif

# plot
plot(enm.tif)

# importar o maxent results
res <- read.csv("maxentResults.csv")
head(res)

# valores dos limites de corte - o numero 11 e para 10 replicas
thr.min <- res[11, colnames(res) == "Minimum.training.presence.logistic.threshold"]
thr.min

thr.10 <- res[11, colnames(res) == "X10.percentile.training.presence.logistic.threshold"]
thr.10

thr.max <- res[11, colnames(res) == "Maximum.test.sensitivity.plus.specificity.logistic.threshold"]
thr.max

# realizando os cortes e a multiplicacao
enm.tif.thr.min <- enm.tif >= thr.min
enm.tif.min <- enm.tif.thr.min * enm.tif

enm.tif.thr.10 <- enm.tif >= thr.10
enm.tif.10 <- enm.tif.thr.10 * enm.tif

enm.tif.thr.max <- enm.tif >= thr.max
enm.tif.max <- enm.tif.thr.max * enm.tif

# lpt - lowest presence threshold
# diretorio dos arquivos vetor
setwd("")

# importar shape de pontos de ocorrencia
pto <- shapefile("bromelia_balansae.shp")
```

```
plot(pto)

# lowest presence threshold
lpt <- min(extract(enm.tif, pto))
lpt

enm.tif.thr.lpt <- enm.tif >= lpt
enm.tif.lpt <- enm.tif.thr.lpt * enm.tif

# plot
par(mfrow = c(2, 2))
plot(enm.tif.lpt, main = paste0("LPT = ", round(lpt, 4)))
plot(enm.tif.min, main = paste0("Min = ", thr.min))
plot(enm.tif.10, main = paste0("10 perc = ", thr.10))
plot(enm.tif.max, main = paste0("Max = ", thr.max))

# stack dos limites de corte
enms.thr.tif <- stack(enm.tif.lpt, enm.tif.min, enm.tif.10, enm.tif.max)

names(enms.thr.tif) <- c("lpt", "min", "10", "max")

# exportar os modelos finais
# diretorio de saida
setwd("")

list <- names(enms.thr.tif)
list

for(i in 1:length(list)){
  writeRaster(enms.thr.tif[[i]], paste0("enm_", list[[i]], ".tif"), format =
"GTiff")}
```

3.4 Histograma

O histograma é um dos gráficos mais simples e que nos fornecem uma boa ideia da distribuição dos valores de adequabilidade do modelo. Esse gráfico relaciona classes de adequabilidade no eixo horizontal e suas respectivas frequências no eixo vertical.

Irei utilizar o mapa de saída direta do software MaxEnt, sem nenhum “corte”, no formato .asc.

Tutorial de Análises Preliminares e Gráficas no R para Modelos de Nicho Ecológico

```
### 3.4 histograma

# diretorio com a saida do maxent
setwd("")

# importar os dados
enm <- raster("B.balansae_avg.asc")

# plot
plot(enm)

# extrair os valores de todos os pixels
enm.v <- values(enm)
head(enm.v)

# retirar os NAs
enm.v.na <- na.omit(enm.v)
head(enm.v.na)

# histograma
tiff("hist.tif", width = 18, height = 18, units = "cm", res = 300, compression =
"lzw")
hist(enm.v.na,                               # dados
     xlim = c(0, 1),                         # limite do eixo x
     br = 10,                                 # numero de classes
     col = "gray",                            # cor das barras
     main = "Frequência da adequabilidade",   # titulo do grafico
     xlab = "Adequabilidade",                # nome do eixo x
     ylab = "",                               # nome do eixo y
     border = "grey40",                      # cor das bordas das colunas
     col.axis = "grey30",                    # cor do texto dos eixos
     cex.lab = 1.5,                           # tamanho do nome dos eixos
     cex.main = 1.5,                          # tamanho do nome do titulo
     cex.axis = 1.2)                          # tamanho dos numeros do eixo
mtext("Frequência", side = 2, line = 2.6, cex = 1.5)
dev.off()
```

3.5 Gráfico de dispersão

Na última seção desse tutorial, vou tratar da parte que acho mais interessante: verificar como os valores de cada pixel do ENM se relacionam com os valores dos pixels de outra variável.

Nesse exemplo, utilizarei os valores da adequabilidade do modelo .asc, do desvio padrão do modelo, da variável Bio01 (Temperatura média anual) e Bio12 (Precipitação anual), e os respectivos valores para os pontos de ocorrência.

```
### 3.5 grafico de dispersao
# 1 adequabilidade e seu desvio padrao
# diretorio com a saida do maxent
setwd("")

# importar os dados
enm <- raster("B.balansae_avg.asc")
enm.df <- raster("B.balansae_stddev.asc")

# plot
plot(enm)
plot(enm.df)

# extrair os valores de todos os pixels
enm.v <- values(enm)
enm.df.v <- values(enm.df)
head(enm.v)
head(enm.df.v)

# retirar os NAs
enm.v.na <- na.omit(enm.v)
enm.df.v.na <- na.omit(enm.df.v)
head(enm.v.na)
head(enm.df.v.na)

# grafico
tiff("dispersao_desvio.tif", width = 18, height = 18, units = "cm", res = 300,
compression = "lzw")
plot(enm.df.v.na ~ enm.v.na, # variaveis
      ylab = "", # nome do y
```


Tutorial de Análises Preliminares e Gráficas no R para Modelos de Nicho Ecológico

```
xlab = "Adequabilidade", # nome do x
main = "Relação entre adequabilidade e seu desvio padrão", # titulo
pch = 20, # tipo de ponto
cex = 1, # tamanho do ponto
col.axis = "grey30", # cor do eixo
cex.lab = 1.5, # tamanho da fonte do nome dos eixos
cex.main = 1.2, # tamanho da fonte do nome do titulo
cex.axis = 1.2, # tamanho dos numeros do eixo
bty = "l") # tipo caixa em volta do grafico
mtext("Desvio padrão da Adequabilidade", side = 2, line = 2.6, cex = 1.5)
dev.off()

# 2 adequabilidade e temperatura e precipitacao anual
# diretorio com os arquivos raster .tif
setwd("")

# importar os dados
bio01 <- raster("bio01_neotropic_50km_gcs_wgs84.asc")
bio12 <- raster("bio12_neotropic_50km_gcs_wgs84.asc")

# plot
plot(bio01)
plot(bio12)

# extrair os valores de todos os pixels
bio01.v <- values(bio01)
bio12.v <- values(bio12)
head(bio01.v)
head(bio12.v)

# retirar os NAs
bio01.v.na <- na.omit(bio01.v)
bio12.v.na <- na.omit(bio12.v)
head(bio01.v.na)
head(bio12.v.na)

# shape de pontos
# diretorio dos arquivos vetor
```

Tutorial de Análises Preliminares e Gráficas no R para Modelos de Nicho Ecológico

```
setwd("")

# importar shape de pontos de ocorrencia
pto <- shapefile("bromelia_balansae.shp")
plot(pto)

# retirar os valores para os pontos
pto.enm <- extract(enm, pto)
pto.enm

pto.bio01 <- extract(bio01, pto)
pto.bio01

pto.bio12 <- extract(bio12, pto)
pto.bio12

# graficos
tiff("dispersao_bio01.tif", width = 18, height = 18, units = "cm", res = 300,
compression = "lzw")
plot(enm.v.na ~ bio01.v.na, # variaveis
      xlim = c(0, 30), # limite do eixo x
      ylab = "", # nome do y
      xlab = "Bio 01 - Temperatura média anual", # nome do x
      main = "Relação entre Adequabilidade e Temperatura", # titulo
      pch = 20, # tipo de ponto
      cex = 1, # tamanho do ponto
      col.axis = "grey30", # cor do eixo
      cex.lab = 1.5, # tamanho da fonte do nome dos eixos
      cex.main = 1.2, # tamanho da fonte do nome do titulo
      cex.axis = 1.2, # tamanho dos numeros do eixo
      bty = "n") # tipo caixa em volta
points(pto.enm ~ pto.bio01, # variaveis
        pch = 20, # tipo de ponto
        cex = 2, # tamanho do ponto
        col = "red") # cor do ponto
mtext("Adequabilidade", side = 2, line = 2.6, cex = 1.5)
dev.off()
```

Tutorial de Análises Preliminares e Gráficas no R para Modelos de Nicho Ecológico

```
tiff("dispersao_bio12.tif", width = 18, height = 18, units = "cm", res = 300,
compression = "lzw")
plot(enm.v.na ~ bio12.v.na,                                     # variaveis
      xlim = c(0, 6000),                                       # limite do eixo x
      ylab = "",                                                # nome do y
      xlab = "Bio 12 - Precipitação anual",                    # nome do x
      main = "Relação entre Adequabilidade e Precipitação", # titulo
      pch = 20,                                                 # tipo de ponto
      cex = 1,                                                  # tamanho do ponto
      col.axis = "grey30",                                       # cor do eixo
      cex.lab = 1.5,                                           # tamanho da fonte do nome dos eixos
      cex.main = 1.2,                                          # tamanho da fonte do nome do titulo
      cex.axis = 1.2,                                          # tamanho dos numeros do eixo
      bty = "l")
points(pto.enm ~ pto.bio12,                                    # variaveis
       pch = 20,                                              # tipo de ponto
       cex = 1.4,                                            # tamanho do ponto
       col = "lightblue")                                     # cor do ponto
mtext("Adequabilidade", side = 2, line = 2.6, cex = 1.5)
dev.off()
```

O pacote “ggplot2” faz gráficos mais elegantes, apesar de ser um pouco mais difícil de utilizar e com uma sintaxe própria, um pouco diferente do padrão do R.

```
# ggplot2

# organizar os dados
dados <- data.frame(ade = enm.v.na, des = enm.df.v.na)
head(dados)

# graficos
ggplot(dados, aes(x = ade, y = des)) +
  geom_point(colour = alpha("black", 0.25)) +
  stat_smooth(aes(x = ade, y = des), colour = "blue") +
  ylab("Desvio Padrão da Adequabilidade") +
  xlab("Adequabilidade") +
  theme(axis.text.x = element_text(colour = "grey20", size = 12, angle = 0, hjust
    = .5, vjust = .5, face = "plain"),
```

Tutorial de Análises Preliminares e Gráficas no R para Modelos de Nicho Ecológico

```
axis.text.y = element_text(colour = "grey20", size = 12, angle = 0, hjust = 1, vjust = 0, face = "plain"),  
axis.title.x = element_text(colour = "grey20", size = 17, angle = 0, hjust = .5, vjust = 0, face = "plain"),  
axis.title.y = element_text(colour = "grey20", size = 17, angle = 90, hjust = .5, vjust = .5, face = "plain"))
```

```
ggsave("dispersao_desvio_gg.tif", width = 18, height = 18, un = "cm", dpi = 300)
```

Agradecimentos

Gostaria primeiramente de agradecer ao Dudu (meu filho) e a Japa (minha esposa) que me aturaram essas semanas que trabalhei nesse tutorial e minhas maiores inspirações. Ao Prof. Milton Cesar Ribeiro, orientador e grande incentivador de ideias e ao seu famoso “Você tem que eRar mais...”. À Profa. Katia M. P. M. de Barros Ferraz, por me apresentar o encantador, desafiador e cheio de incertezas mundo da modelagem de nicho. Ao Prof. Thadeu Sobral-Souza, por todas as conversas, incentivos e desafios que me propôs, fazendo com que eu me supera-se, esse tutorial é parte dessa superação. Por fim, mas nem por isso menos importante, ao Prof. Tadeu Siqueira de Barros, com quem fiz meu primeiro curso de R.



(Eu e o Dudu)

Referências

Barve, N., Barve, V., Jiménez-Valverde, A., et al., 2011. The crucial role of the accessible area in ecological niche modeling and species distribution modeling. *Ecol. Model.* 222, 1810-1819.

Hijmans, R.J., S.E. Cameron, J.L. Parra, P.G. Jones and A. Jarvis, 2005. Very high resolution interpolated climate surfaces for global land areas. *International Journal of Climatology* 25: 1965-1978.

- Lima-Ribeiro MS; Varela S; González-Hernández J; Oliveira G; Diniz-Filho JAF; Terribile LC. (2015) ecoClimate: a database of climate data from multiple models for past, present, and future for Macroecologists and Biogeographers. *Biodiversity Informatics* 10, 1-21.
- Lima-Ribeiro MS; Varela S; González-Hernández J; Oliveira G; Diniz-Filho JAF; Peterson AT; Terribile LC. The ecoClimate Database, <http://ecoclimate.org>, accessed <dez 09, 2016>.
- Peterson, A. Townsend, Jorge Soberón, Richard G. Pearson, Robert P. Anderson, Enrique Martínez-Meyer, Miguel Nakamura, and Miguel Bastos Araújo. *Ecological Niches and Geographic Distributions (MPB-49)*. Princeton University Press, 2011. <http://www.jstor.org/stable/j.ctt7stnh>.
- R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Sobral-Souza, Thadeu; Lima-Ribeiro, Matheus S.; Solferini, Vera Nisaka. Biogeography of Neotropical Rainforests: past connections between Amazon and Atlantic Forest detected by ecological niche modeling. *Evolutionary Ecology*, v. 29, p. 1, 2015.
- Steven J. Phillips, Miroslav Dudík, Robert E. Schapire. A maximum entropy approach to species distribution modeling. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 655-662, 2004.