Tree species delimitation in tropical forest inventories: perspectives from a taxonomically challenging case study

Paulo Henrique Gaem[a,*], Ana Andrade [b], Fiorella Fernanda Mazine [a] & Alberto Vicentini [c]

[a] Grupo de Pesquisa em Ecologia, Sistemática e Conservação de Recursos Naturais, Universidade Federal de São Carlos, Campus Sorocaba, Rodovia João Leme dos Santos (SP-264), km 110, 18052-780, Sorocaba, SP, Brazil

[b] Projeto Dinâmica Biológica de Fragmentos Florestais, Instituto Nacional de Pesquisas da Amazônia, Avenida André Araújo, 2936, 69067-375, Manaus, AM, Brazil

[c] Laboratório de Botânica Amazônica, Programa de Pós-Graduação em Botânica, Instituto Nacional de Pesquisas da Amazônia, Avenida André Araújo, 2936, 69067-375, Manaus, AM, Brazil

[*] Corresponding author

Supplementary File 3. Programming routine used to data preparation and statistical analyses in R environment.

```
#Tested classifications
classificacoes =  c("class_paulo", "class_aleatoria")
#Dataset
file=read.table(‘file.csv’, sep=‘/t’, as.is=T, header=T);
rownames(file)=file$id_1
```

1. Stepwise procedure with spectral data

```
#Selection of data used in stepwise procedure
identifier=file$id_2, names=file$morph_class, nirdad=file[,16:1572]
espectros=cbind(identifier, names, nirdad)
#Remove species with one sample only
library(dplyr)
especs_once=count(espectros, espectros$names)
```

```
especs_once=subset(especs_once, especs_once$n==1)

especs_more=espectros[!espectros$names %in% especs_once$`espectros$names`,]

detach("package:dplyr", unload=TRUE)

#Stepwise procedure itself

library(MASS)

library(klaR)

x=as.matrix(nirdad); grouping=names; method=‘lda’; direction=‘both’;  criterion=‘CR’; fold=10;

maxvar=floor(nrow(x)/3);

stepwise.res = stepclass(x, grouping, method, maxvar = maxvar, direction = direction, criterion =

criterion, fold = fold,  output = F,  min1var = TRUE)

formula=sc_obj$formula

vars=all.vars(formula)

vars=vars[-1]

nirsel=nirdad[,(names(nirdad) %in% formula)]
```

2. PCA-reduction of spectral data

```
nirpca = prcomp(nirdad)

nirpca = nirpca$x[,1:129]

nirpca = as.data.frame(nirpca,stringsAsFactors = F)

rownames(nirpca) = rownames(nirdad)
```

3. Morphometric data

```
morph=file[,5:15]
```

4. Spectral plus morphometric datasets

```
rn = rownames(morph)

nir_morph = cbind(nirdad[rn,],morph)

nirsel_morph = cbind(nirsel[rn,],morph)

rn = rownames(morph)
```

```
nirpcamorph = cbind(nirpca[rn,],morph)
```

5. Define a leave-one-out cross-validation LDA

```
library("caret")

myControl_LOOCV <- trainControl(
      method = "LOOCV",
      ## outras definicoes
      # imprime iteracao na tela?
      verboseIter = TRUE,
      # salva as predicoes
      savePredictions = TRUE,
      # salva a probabilidade das classes
      classProb = TRUE
)
```

6. Calculate the models’ accuracies for the morphotype-based classification and the aleatory classification

```
resultado.logcv = NULL
cl=1
for(cl in 1:length(classificacoes)) {
  set.seed(453)

  classe = classificacoes[cl]

  ln = rownames(nirdad)
  grouping = as.vector(nomes_morph[ln,classe])
  tbl = table(grouping)
  tira = which(grouping%in%names(tbl[tbl<3]))
  print(paste(classe,"teste NIR tirado as especies:",paste(grouping[tira],collapse = " ")))
  lda1 <- train(x=nirdad[-tira,],y=grouping[-tira],method="lda",trControl = myControl_LOOCV)
  n2 = paste(classe,"nirdad",sep= ".")
  print(paste(n2,"ACCURACY: ",round(lda1$results$Accuracy,2)*100,"%",sep=""))
  resultado.logcv[[n2]] = lda1

  ln = rownames(morph)
  grouping = as.vector(nomes_morph[ln,classe])
  tbl = table(grouping)
  tira = which(grouping%in%names(tbl[tbl<3]))
  print(paste(classe,"teste morph tirado as especies:",paste(grouping[tira],collapse = " ")))
  lda2 <- train(x=morph[-tira,],y=grouping[-tira],method="lda",trControl = myControl_LOOCV)
  n2 = paste(classe,"morph",sep= ".")
  print(paste(n2,"ACCURACY: ",round(lda2$results$Accuracy,2)*100,"%",sep=""))
  resultado.logcv[[n2]] = lda2

  ln = rownames(nir_morph)
  grouping = as.vector(nomes_morph[ln,classe])
  tbl = table(grouping)
  tira = which(grouping%in%names(tbl[tbl<3]))
  print(paste(classe,"teste nir+morph tirado as especies:",paste(grouping[tira],collapse = " ")))
  lda3 <- train(x=nir_morph[-tira,],y=grouping[-tira],method="lda",trControl =
myControl_LOOCV)
  n2 = paste(classe,"nir+morph",sep= ".")
  print(paste(n2,"ACCURACY: ",round(lda3$results$Accuracy,2)*100,"%",sep=""))
  resultado.logcv[[n2]] = lda3
  ln = rownames(nirsel)

  grouping = as.vector(nomes_morph[ln,classe])

  tbl = table(grouping)

  tira = which(grouping%in%names(tbl[tbl<3]))

  print(paste(classe,"teste NIRsel tirado as especies:",paste(grouping[tira],collapse = " ")))

  lda4 <- train(x=nirsel[-tira,],y=grouping[-tira],method="lda",trControl = myControl_LOOCV)

  n2 = paste(classe,"nirsel",sep= ".")

  print(paste(n2,"ACCURACY: ",round(lda4$results$Accuracy,2)*100,"%",sep=""))

  resultado.logcv[[n2]] = lda4


  ln = rownames(nirsel_morph)

  grouping = as.vector(nomes_morph[ln,classe])

  tbl = table(grouping)

  tira = which(grouping%in%names(tbl[tbl<3]))

  print(paste(classe,"teste nirsel+morph tirado as especies:",paste(grouping[tira],collapse = " ")))

  lda5 <- train(x=nirsel_morph[-tira,],y=grouping[-tira],method="lda",trControl =
myControl_LOOCV)

  n2 = paste(classe,"nirsel+morph",sep= ".")

  print(paste(n2,"ACCURACY: ",round(lda5$results$Accuracy,2)*100,"%",sep=""))

  resultado.logcv[[n2]] = lda5


  ln = rownames(nirpca)

  grouping = as.vector(nomes_morph[ln,classe])

  tbl = table(grouping)

  tira = which(grouping%in%names(tbl[tbl<3]))
  print(paste(classe,"teste nirPCA tirado as especies:",paste(grouping[tira],collapse = " ")))
  lda6 <- train(x=nirpca[-tira,],y=grouping[-tira],method="lda",trControl = myControl_LOOCV)
  n2 = paste(classe,"nirPCA",sep= ".")
  print(paste(n2,"ACCURACY: ",round(lda6$results$Accuracy,2)*100,"%",sep=""))
  resultado.logcv[[n2]] = lda6

  ln = rownames(nirpcamorph)
  grouping = as.vector(nomes_morph[ln,classe])
  tbl = table(grouping)
  tira = which(grouping%in%names(tbl[tbl<3]))
  print(paste(classe,"teste nirPCA+morph tirado as especies:",paste(grouping[tira],collapse = " ")))
  lda7 <- train(x=nirpcamorph[-tira,],y=grouping[-tira],method="lda",trControl =
myControl_LOOCV)
  n2 = paste(classe,"nirPCA+morph",sep= ".")
  print(paste(n2,"ACCURACY: ",round(lda7$results$Accuracy,2)*100,"%",sep=""))
  resultado.logcv[[n2]] = lda7
}
```

7. Save Rdata

```
save(resultado.logcv,file="testemorfotipo.Rdata")
```

8. Calculate accuracies

```
load("testemorfotipo_indet16.Rdata")
pegaaccuracia <- function(x) {
 round(x$results$Accuracy,2)
}
acuracias <- sapply(resultado.logcv,pegaaccuracia)
names(acuracias)=names(resultado.logcv)
```

```
acuracias = as.data.frame(acuracias,stringsAsFactors=F)

colnames(acuracias) = "Acuracia"

acuracias

write.table(acuracias,sep="\t",na="",quote=T,row.names = T)
```

9. Confusion matrixes

```
load("testemorfotipo.Rdata")

abc=resultado.logcv$'classification'$pred[,1:2]

plotamatriz <- function(matriz,bg.cols,txt.cols, valcex=1, cexaxis=1) {

  tb = matriz

  xx = 1:ncol(tb)

  yy = seq(1,nrow(tb),length.out=length(xx))

  plot(xx,yy,type='n',xaxt='n',yaxt='n',xlab='',ylab='', pty='m',

xlim=c(0.5,max(xx)+.5),ylim=c(0.5,max(yy)+0.5))

  ay = 1:ncol(tb)

  ax = 1:nrow(tb)

  abline(v=ay,h=ax,lty='dotted',lwd=0.5)

  nn = rownames(tb)

  nncl = colnames(tb)

  for(l in 1:nrow(tb)) {

    for(cc in 1:ncol(tb)) {

      cl = bg.cols[l,cc]

      txtc = txt.cols[l,cc]

      val = tb[l,cc]

      if (!is.na(val)) {

        rect(xleft=cc-0.5,ybottom=l-0.5,xright=cc+0.5,ytop=l+0.5,density=-1,border=NA,col=cl)

        text(cc,l,labels=val,cex=valcex,col=txtc)
      }
    }
  }
  axis(side=3,at=ay,labels=colnames(tb),cex.axis=cexaxis,las=2)
  axis(side=2,at=ax,labels=rownames(tb),cex.axis=cexaxis,las=2)
}
tb = table(abc$obs,abc$pred)
tb = as.matrix(tb)
cl = rownames(tb)
cl2 = colnames(tb)
falta = cl[!cl%in%cl2]
if (length(falta)>0) {
  mm = matrix(0,nrow=nrow(tb),ncol=length(falta),dimnames = list(rownames(tb),falta))
  tb = cbind(tb,mm)
  rn  = sort(rownames(tb),decreasing = F)
  rn2 = sort(rownames(tb),decreasing = F)
  tb = tb[rn,rn2]
}
bg.cols= tb
txt.cols = tb
unique(as.vector(tb))
bg.cols[tb==0] = gray(level=1)
txt.cols[tb==0] = gray(level=1)
bg.cols[tb>0 & tb<=2] = gray(level=0)
txt.cols[tb>0 & tb<=2] = gray(level=1)
bg.cols[tb>2] = gray(level=0)
```

```
txt.cols[tb>2] = gray(level=1)

tb[tb==0] = NA

fn = paste("confusion_matrix.png")

png(file=fn, width = 21, height = 21, units = "cm", pointsize = 10,bg = "white",  res = 600)

par(mar=c(2,10,10,2))

plotamatriz(tb,bg.cols=bg.cols,txt.cols=txt.cols,valcex=0.8, cexaxis = 0.8)

dev.off()
```