```
#########################################################################
# TRABAJO DE FIN DE GRADO
# Tratamiento de datos químico-forenses para la discriminación
# de fluidos biológicos en materiales superabsorbentes
#########################################################################
# Autor: Ignacio Pachón Jiménez #
#LOAD PACKAGES#
library("ChemoSpec")
library("R.utils")
library("baseline")
library("IDPmisc")
library("signal")
library("stats")
library("Hmisc")
library("graphics")
library("ROCR")
library("OptimalCutpoints")
#LOAD SPECTRAL DATA#
#Read the Dataset
files2SpectraObject(gr.crit=c("Blank","Mixture","Semen","Urine","Vagin
al fluid"),
gr.cols=c("red3", "dodgerblue4", "forestgreen", "purple4",
"orangered4"),freq.unit="",
int.unit="",descrip="Fluidos biológicos en materiales
absorbentes",out.file="1 TFGSpectra")
xaxis<-expression(cm^-1)
yaxis<-expression(Log (1/R))
#Load de Dataset



All <- loadObject("1 TFGSpectra.RData")
########################
#LOAD SPECTRAL DATA TRANFORMATION FUNCTIONS#
```

```
# Load custom normalization function (normNacho)
normNacho <- function(spectra) {
# Function to normalize a Spectra object so that each spectrum
# is on a [0...1] scale
# Bryan Hanson, DePauw University, Feb 2016
if (missing(spectra)) stop("No spectral data provided")
chkSpectra(spectra)
for (i in 1:length(spectra$names)) {
rMin <- min(spectra$data[i,])
spectra$data[i,] <- spectra$data[i,] - rMin
rMax <- max(spectra$data[i,])
spectra$data[i,] <- spectra$data[i,]/rMax
}
chkSpectra(spectra)
return(spectra)
}
# Load custom Smoothing function. Savitzky-Golay
sgfSpectra <- function(spectra, m = 0) {
# Function to filter a Spectra object
# Bryan Hanson, DePauw University, Feb 2016

if (!requireNamespace("signal", quietly = TRUE)) {
stop("You need to install package signal to use this function")
}
if (missing(spectra)) stop("No spectral data provided")
chkSpectra(spectra)
for (i in 1:length(spectra$names)) {
spectra$data[i,] <- sgolayfilt(spectra$data[i,],p=2,n=11,m=0)
}
chkSpectra(spectra)
return(spectra)
}
```

```
# Load custom baseline correction function (baselineNacho)
baselineNacho <- function(spectra) {
# Bryan Hanson, DePauw University, Feb 2016
if (missing(spectra)) stop("No spectral data provided")
chkSpectra(spectra)
np <- length(spectra$freq)
for (i in 1:length(spectra$names)) {
rMin <- min(spectra$data[i,])
spectra$data[i,] <- spectra$data[i,] - rMin
# Do an lm from end to the other
DF <- data.frame(
x = c(spectra$freq[1], spectra$freq[np]),
y = c(spectra$data[i,1], spectra$data[i,np]))

fit <- lm(y ~ x, DF)
spectra$data[i,] <- spectra$data[i,]- predict(fit,
newdata = data.frame(x = spectra$freq))
}
chkSpectra(spectra)
return(spectra)
}
#PRELIMINARY INSPECTION OF DATA#
################################
sumSpectra(All)
#DATA PRE-PROCESSING#
## Remove frequencies. Selecting research's Range
Ranged<-removeFreq(All,rem.freq=All$freq>1690|All$freq<1500)
meanRAllsd<-surveySpectra(Ranged,method="sd",main="Media de espectros
+/- desviación estandar")
#BASELINE CORRECTION#
#Baseline offset f(x)=x-min(X)--->baselineNacho
#Linear Baseline Correction.
```

```
BsRanged<-baselineNacho(Ranged)

#SMOOTHING#

SmBsRanged<-sgfSpectra(BsRanged)

#NORMALIZATION#

NSBRanged<-normNacho(SmBsRanged)



######################

meanNSBRAllsd<-surveySpectra(NSBRanged,method="sd",main="Media de
espectros +/- desviación estandar")

# PCA (Análisis de Componentes Principales) #

PCA_NSBR <- c_pcaSpectra(NSBRanged, choice = "noscale")

plotScores(NSBRanged,main="Scores PCA con
Blancos",PCA_NSBR,pcs=c(1,2))

diagnosticsOD <- pcaDiag(NSBRanged, PCA_NSBR, pcs = 10, plot = "OD")

diagnosticsSD <- pcaDiag(NSBRanged, PCA_NSBR, pcs = 5, plot = "SD")

plotScoresRGL(NSBRanged, PCA_NSBR,leg.pos = "A",t.pos = "B")

plotScores3D(NSBRanged, PCA_NSBR, main = title, ellipse = T)

plotLoadings(NSBRanged, PCA_NSBR, main = title,loads = c(1,2,3),ref=1)

########################

NSBRPuros<-removeGroup(NSBRanged,"Blank")

Puros_PCA_NSBR <- c_pcaSpectra(NSBRPuros, choice = "noscale")

plotScores(NSBRPuros,main="Scores sin
Blancos",Puros_PCA_NSBR,pcs=c(1,2))

diagnosticsOD <- pcaDiag(NSBRPuros, Puros_PCA_NSBR, pcs = 10, plot =
"OD")

diagnosticsSD <- pcaDiag(NSBRPuros, Puros_PCA_NSBR, pcs = 5, plot =
"SD")

plotScoresRGL(NSBRPuros, Puros_PCA_NSBR,leg.pos = "A",t.pos = "B")

plotScores3D(NSBRPuros, Puros_PCA_NSBR, main = title, ellipse = T)

plotLoadings(NSBRPuros, Puros_PCA_NSBR, main = title,loads =
c(1,2,3),ref=1)
```

```
###### WARNING!!!! Set a different directory (not a database) #######
# PEARSON (r) #
# Cargar funciones para los Coef Corr Inter e Intra
#cor.test {stats}
cor.testInter <- function(x,y){
FUN <- function(x, y) cor.test(x, y)[["estimate"]]
z <- outer(
colnames(x),
colnames(y),
Vectorize(function(i,j) FUN(x[,i], y[,j]))

)
dimnames(z) <- list(colnames(x), colnames(y))
z
}
cor.testIntra <- function(x){
FUN <- function(x, y) cor.test(x, y)[["estimate"]]
z <- outer(
colnames(x),
colnames(x),
Vectorize(function(i,j) FUN(x[,i], x[,j]))
)
dimnames(z) <- list(colnames(x), colnames(x))
z
}
# Export processed spectra
#Spectra must be columns, NOT ROWS!
#Blank== 1:170
#Mix== 171:250
#Sem== 251:303
#Uri== 304:361
#Vag== 362:406
```

```
write.table(t(NSBRanged$data),file="PearsonMatrix.csv",
quote=F,sep=";",dec=",",row.names=F,col.names=F)
#Once created the table, proceed to import data
#Spectra are still cols.
pearsonMatrix<-read.csv("PearsonMatrix.csv",header=F,sep=";",dec=",")

#Check these spectra are the same
plot(pearsonMatrix$V1,type="l")
plotSpectra(NSBRanged,which=c(1))
## LET'S DEFINE SOME POPULATIONS
# Populations to correlate
# All
dfAll<-as.data.frame(pearsonMatrix)
# Semen
dfSemen<-as.data.frame(dfAll[,251:303])
# No Semen (Vaginal Fluid and Urine)
dfNoSemen<-as.data.frame(dfAll[,304:406])
# Mixes
#227 +++
#217 ---
#192 -- (Scenario 0, o Scenario 2 Alternative)
dfMezclasEscenario1<-as.data.frame(dfAll[,227])
dfMezclasEscenario2<-as.data.frame(dfAll[,217])
dfMezclas<-cbind(dfMezclasEscenario1,dfMezclasEscenario2)
#dfMezclasEscenario0<-as.data.frame(dfAll[,192])
#227 +++Intensity (Scenario 1)
#217 ---Intensity (Scenario 2)
#192 --Intensity (Scenario 0, or Scenario 2 Alternative)
dfAll<-as.data.frame(pearsonMatrix)
dfSemen<-as.data.frame(dfAll[,251:303])
dfNoSemen<-as.data.frame(dfAll[,304:406])
dfMezclasEscenario1<-as.data.frame(dfAll[,227])
```

```
dfMezclasEscenario2<-as.data.frame(dfAll[,217])

dfMezclas<-cbind(dfMezclasEscenario1,dfMezclasEscenario2)

#dfMezclasEscenario0<-as.data.frame(dfAll[,192])


rIntraSemen<-cor.testIntra(dfSemen)

rInter<-cor.testInter(dfSemen,dfNoSemen)

rInterM1<-cor.testInter(dfSemen,dfMezclasEscenario1)

rInterM2<-cor.testInter(dfSemen,dfMezclasEscenario2)

###################

range(rInter)

range(rInterM)

range(rIntraSemen)

#PLOT HISTOGRAMS#

histIntraSemen<-
hist(rIntraSemen,freq=F,col="green",main="Intravariabilidad Semen vs
Intervariabilidad",border="green",breaks=90,xlim=c(-
0.86,1),ylim=c(0,35),add=F)

histInterM1<-
hist(rInterM1,freq=F,col="purple",border="purple",main="Intervariabili
dad Escenario 1",breaks=50)

histInterM2<-
hist(rInterM2,freq=F,col="purple",border="purple",main="Intervariabili
dad Escenario 2",breaks=50,ylim=c(0,35),xlim=c(0.73,1))

histInter<-
hist(rInter,freq=F,col="red",border="red",main="Intervariabilidad
Semen vs. No Semen",breaks=555)

# 1.Inter vs Intra

plot(histIntraSemen,col=rgb(1,0.4,0,1/2),axes=F,border=rgb(1,0.4,0,1/2
),freq=F,xlab="Coeficientes de Correlación de
Pearson",ylab="Frecuencia relativa (%)",main="Inter vs. Intra")

plot(histInter,col=rgb(0.5,1,0,1/2),axes=F,border=rgb(0.5,1,0,1),freq=
F,add=T,xlab="Coeficientes de Correlación de Pearson",ylab="Frecuencia
relativa (%)")

legend(0.8,14,bty="n",legend=c("Intra (Semen)","Inter (Semen vs No
Semen)"),

text.col="black",fill=c(rgb(1,0.4,0,1/2),rgb(0.5,1,0,1/2)))

axis(1,at=seq(0.5,1,by=0.5),labels=seq(0.5,1,by=0.5))
```

```
axis(1,at=seq(0.5,1,by=0.05),labels=seq(0.5,1,by=0.05))

axis(2,at=seq(0,24,by=2),las=1)

lines(density(rIntraSemen),col="orangered",lwd=3)

lines(density(rInter),col="green",lwd=3)

# 2.Scenario 1


plot(histIntraSemen,col=rgb(1,0.4,0,1/2),border=rgb(1,0.4,0,1),freq=F,
axes=F,add=F,xlab="Coeficientes de Correlación de
Pearson",ylab="Frecuencia
relativa(%)",xlim=c(0.75,1),ylim=c(0,35),main="Escenario 1")

plot(histInter,col=rgb(0.5,1,0,1/2),border=rgb(0.5,1,0,1),freq=F,add=T
,axes=F,xlab="Coeficientes de Correlación de Pearson",ylab="Frecuencia
relativa(%)")

plot(histInterM1,col=rgb(0.37,0.07,0.56,1/2),border=rgb(0.37,0.07,0.56
,1),axes=F,freq=F,add=T,xlab="Coeficientes de Correlación de
Pearson",ylab="Frecuencia relativa(%)")

legend(0.85,26,bty="n",legend=c("Intra (Semen)","Inter (Semen vs No
Semen)","Inter (Semen vs Mezcla
1)"),fill=c(rgb(1,0.4,0,1/2),rgb(0.5,1,0,1/2),rgb(0.37,0.07,0.56,1/2))
)

lines(density(rIntraSemen),col="orangered",lwd=3)

lines(density(rInter),col="green",lwd=3)

lines(density(rInterM1),col="purple",lwd=3)

axis(1,at=seq(0.75,1,by=0.005),labels=seq(0.75,1,by=0.005))

axis(2,at=seq(0,35,by=2),las=1)

# 3.Scenario 2

plot(histInter,axes=F,col=rgb(0.5,1,0,1/2),border=rgb(0.5,1,0,1),freq=
F,add=F,xlab="Coeficientes de Correlación de Pearson",ylab="Frecuencia
relativa(%)",main="Escenario 2",xlim=c(0.75,1),ylim=c(0,35))

plot(histIntraSemen,axes=F,col=rgb(1,0.4,0,1/2),border=rgb(1,0.4,0,1),
freq=F,add=T,xlab="Coeficientes de Correlación de
Pearson",ylab="Frecuencia relativa(%)")

plot(histInterM2,axes=F,add=T,col=rgb(0.37,0.07,0.56,1/2),border=rgb(0
.37,0.07,0.56,1),freq=F,xlab="Coeficientes de Correlación de
Pearson",ylab="Frecuencia relativa(%)")

legend(0.75,18.5,bty="n",legend=c("Intra (Semen)","Inter (Semen vs No
Semen)","Inter (Semen vs Mezclas
2)"),fill=c(rgb(1,0.4,0,1/2),rgb(0.5,1,0,1/2),rgb(0.5,0.5,1,1/2)))
```

```
axis(1,at=seq(0.75,1,by=0.005),labels=seq(0.75,1,by=0.005))
axis(2,at=seq(0,35,by=2),las=1)
lines(density(rIntraSemen),col="orangered",lwd=3)
lines(density(rInter),col="green",lwd=3)
lines(density(rInterM2),col="purple",lwd=3)
# ROC & roll #
labIntra<-seq(1,1,length=length(rIntraSemen))
labInter<-seq(0,0,length=length(rInter))
labels<-c(labIntra,labInter)

preds<-c(rIntraSemen,rInter)
pred.obj<-prediction(preds,labels)
tpr<-performance(pred.obj,"tpr")
fpr<-performance(pred.obj,"fpr")
fnr<-performance(pred.obj,"fnr")
tnr<-performance(pred.obj,"tnr")
TP<-as.data.frame(tpr@"y.values")
FP<-as.data.frame(fpr@"y.values")
#PLOT CURVES#
plot(fpr,col="black",ylab="",xlab="",box.lty=0,lwd=5)
plot(tpr,col="green",ylab="",xlab="",add=T,lwd=5)
plot(fnr,col="red",ylab="",xlab="",add=T,lwd=5)
plot(tnr,col="blue",ylab="",xlab="",add=T,lwd=5)
mtext("Ratio",side=2,line=2)
axis(1,at=seq(0,1,by=0.05),labels=F)
axis(1,at=seq(-0.9,1,by=0.1),labels=T)
axis(2,at=seq(0.1,0.9,by=0.2),labels=T)
mtext("Coeficientes de Correlación de Pearson",side=1,line=2)
grid()
legend(-0.49,0.79,bty="",legend=c(" Ratios","Falsos
positivos","Verdaderos positivos","Falsos negativos","Verdaderos
negativos"),
```

```
text.col=c("black","black","green","red","blue"),pch=c("","--","--","-
-","--"),col=c("black","black","green","red","blue"))

ROCcurve<-performance(pred.obj,"tpr","fpr")

ROCcurve

plot(ROCcurve,col="red3",lwd=5,main="Curva ROC")

ROCauc<-performance(pred.obj,"auc")

ROCauc@"y.values"

# Otros cálculos ROC


# AUC

ROCauc<-performance(pred.obj,"auc")

ROCauc@"y.values"

cutpoints.obj<-data.frame(preds,labels)

data<-cutpoints.obj

MaxSpSe<-
optimal.cutpoints(preds~labels,tag.healthy=0,"MaxSpSe",cutpoints.obj)

MaxSp<-
optimal.cutpoints(preds~labels,tag.healthy=0,"MaxSp",cutpoints.obj)

MaxSe<-
optimal.cutpoints(preds~labels,tag.healthy=0,"MaxSe",cutpoints.obj)

Youden<-
optimal.cutpoints(preds~labels,tag.healthy=0,"Youden",cutpoints.obj)

MaxEffi<-
optimal.cutpoints(preds~labels,tag.healthy=0,"MaxEfficiency",cutpoints
.obj)


# CHECK RESULTS!

str(MaxSpSe)

str(MaxSp)

str(MaxSe)

str(Youden)

str(MaxEffi)

#########################################################################
```