Ejemplo práctico Andurinha

Miguel Suárez-Couselo

2022-06-05

## Seteamos el directorio y cargamos las librerias que necesitaremos

setwd("c:/Users/Usuario/Desktop/Andurinha/Andurinha/")  
library(andurinha)

## Cargamos los datos de FTIR

datos <- read.csv("./Resultados\_FTIR.csv", header = T, sep = ";", dec = ",")

# Función findPeaks

### Que hace esta función?

Esta función explora los datos y calcula el valor de segunda derivada para cada uno de los picos. Además nos permite seleccionar un “cutOff” (punto de corte) a partir de un valor de segunda derivada que nosotros seleccionemos y así poder quedarnos solo con los picos mas “relevantes” de nuestras muestras

### Que contiene el objeto que crea esta función?

El objeto que crea esta función contiene 4 sets de datos distintos

1. dataZ: son los valores del FTIR que ya teniamos pero estandarizados mediante Z-scores
2. secondDerivative: los valores de segunda derivada de nuestros datos
3. sumSpectrum\_peaksTable: las longitudes de onda y los valores de segunda derivada que corresponden a cada uno
4. peaksTable: los picos que hemos seleccionado al aplicar el cutOff y sus valores de absorvancia

Aplicamos por lo tanto la función creando un nuevo objeto de datos procesados:

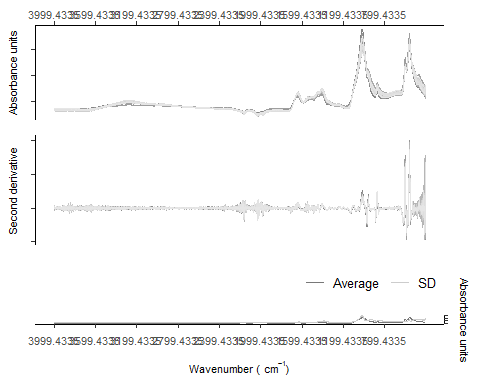
picos <- findPeaks(data = datos)

# Función gOverview

Con esta función vamos a poder ver la gráfica de absorvancia de los datos y además la gráfica de valores de segunda derivada de cada uno de las longitudes de onda. Los argumentos de esta función son:

1. data\_abs: un dataframe con las longitudes de onda en la primera columna y los valores de las muestras en las siguientes columnas (en nuestro caso son los datos que tenemos)
2. data\_ndd: un dataframe que contenta en la primera columna las longitudes de onda y en la segunda los valores de segunda derivada de las muestras. Esta función la utilizaremos solo cuando queramos que en la gráfica se vean tambien los valores de segunda derivada

gOverview(data\_abs = picos$dataZ, data\_ndd = picos$secondDerivative)



En la parte de arriba de la gráfica podemos ver la representación de las absorvancias de cada una de nuestras muestras. En la gráfica del medio vemos los valores de segunda derivada para cada longitud de onda y en la gráfica de abajo del todo la linea mas oscura representa la absorvancia media (calculada a partir de la media de absorvancia de todas las muestras en cada longitud de onda) y la linea mas clara representa la desviación estandar de esa media.

Esta gráica nos permite ver las zonas en las que tenemos los picos en nuestras muestras y por lo tanto caracterizar la composición de esas zonas.

Para poder determinar un cutOff bueno necesitamos ver la tabla de valores de segunda derivada de cada una de las longitudes de onda. Para esto utilizaremos la siguiente función:

View(picos$sumSpectrum\_peaksTable)

Si ordenamos de mayor a menor la columna de sumSpectrum pulsando en su nombre podremos ver que se ve un “salto” en valores de segunda derivada entre los valores de 11 y 9 por lo que quizás podria tratarde de un buen punto para establecer el cutOff

Para establecer dicho cutOff y que seleccione los picos que tengan un valor de segunda derivada mayor de 10 tenemos que aplicaar la función de findPeaks de la siguiente forma:

cutoff <- findPeaks(data = datos, cutOff = 10)

Para poder visualizar que picos ha seleccionado y si nos convence como está vamos a utilizar la función plotPeaks:

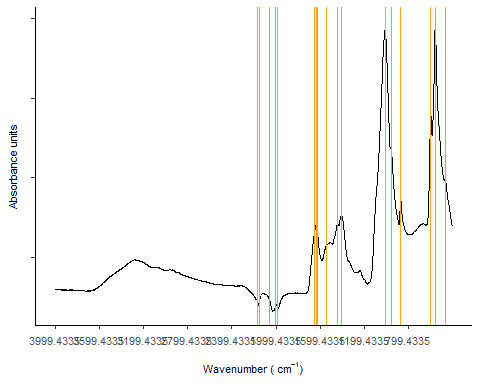
# Función plotPeaks

Esta función va a representar una gráfica similar a la que nos proporciona gOverview solo que en este caso vamos a tener la oportunidad de marcar los picos que han sido seleccionados por el cutOff que hemos aplicado. Los argumentos de esta función son:

1. peaksWN: un vector que contenga los picos que ha seleccionado el cutOff
2. data\_abs: un dataframe que contenga todos nuestros datos (en nuestro caso este dataframe es el objeto “datos”)
3. data\_ndd: un dataframe que contenga todos los valores de segunda derivdad, esto nos representaria una gráfica de segunda derivada pero yo en este caso no lo voy a hacer, aunque se podría

La representación se hace de la siguiente forma:

plotPeaks(peaksWN = cutoff$sumSpectrum\_peaksTable$WN, data\_abs = cutoff$dataZ)



Aquí podemos obervar que nos genera una gráfica con el valor medio de absorvancia de todas nuestras muestras y los picos que selecciona, si consideramos que selecciona pocos picos simplemente habría que repetir el proceso utilizando un cutOff mas bajo y si pensamos que selecciona demasiados picos tendriamos que repetir el proceso utilizando un cutOff mas alto.

Para poder acceder a los picos que ha seleccionado simplemente tenemos que ir al apartado sumSpectrum\_peaksTable del objeto que hayamos creado con el cutOff (en nuestro caso el objeto se llama cutOff) de la siguiente forma:

View(cutoff$sumSpectrum\_peaksTable)

Cualquiera de estas tablas generadas se puede exportar a formato excel o formato csv utilizando las funciones write.xlsx y write.csv respectivamente.