Example\_Processing\_Algorythm\_ShareN.R

gmi415

Wed Aug 28 11:08:15 2019

#This is an implimentation of the deconvolution procedure described in Bowden and Taylor  
#To use the document, open it in R, check directory structure and run  
#clean environment and get packages  
rm(list=ls())  
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.4.4

library(grid)  
library(plyr)

## Warning: package 'plyr' was built under R version 3.4.4

library(DescTools)

## Warning: package 'DescTools' was built under R version 3.4.3

#Normalising, rescaling and end point finding functions  
rescale\_01 <- function(x){  
# Normalizes a vector to [0,1]  
 minr<-min(x)  
 maxr<-max(x)  
 (x - minr) / (maxr - minr)  
}  
#min and max finders  
topMinUsingPartialSort <- function(x, N) {  
 N <- min(N, length(x))  
 x[x <= sort(x, partial=N)[N]][1:N]  
}  
topMaxUsingPartialSort <- function(x, N) {  
 N <- min(N, length(x))  
 x[x >= -sort(-x, partial=N)[N]][1:N]  
}  
  
#name of file to process  
name\_of\_file<-"SP\_37.txt"  
  
#Point at directory containing samples  
setwd("C:/Users/gmi415/Documents/Brough\_deconvolve/Example\_SERS\_DATA")  
  
#Load sample and remove header etc.and data out of range of interest  
#Note that the read table function will have to be adjusted for different file formats  
BR2<-read.table((file=paste(name\_of\_file)), sep = ";", skip= 91)  
BR2n<-data.frame(BR2[,4], BR2[,8])  
colnames(BR2n)<-c("ramnum","V2")  
BR2n<-subset(BR2n, ramnum > 1200| ramnum < 1800)  
BR2n<-subset(BR2n, ramnum >= 1200 & ramnum <= 1800)  
BR2n<-as.data.frame(supsmu(BR2n$ramnum, BR2n$V2, span = "0.03"))  
colnames(BR2n)<-c("ramnum","V2")  
  
#baseline (find end points and substract a straightline)  
minl<-subset(BR2n, V2 == min(BR2n[1:30,2]))   
minh<-subset(BR2n, V2 == min(BR2n[(nrow(BR2n)-20):nrow(BR2n),2]))   
minl[,2]<- mean(topMinUsingPartialSort(BR2n[1:30,2], 10))  
minh[,2]<-mean(topMaxUsingPartialSort(BR2n[(nrow(BR2n)-20):nrow(BR2n),2], 10))  
baslinepoints<-rbind(minl,minh)  
linearMod <- lm(V2 ~ ramnum, data=baslinepoints)  
baseline<-predict(linearMod, BR2n)  
BR2n[,3]<-baseline  
BR2n$Spec<-(BR2n[,2]-baseline)  
BR2n[BR2n<0]<-0  
  
#find height for calibration - compare before and after  
Gband<-(BR2n[which(BR2n$ramnum > 1570 & BR2n$ramnum < 1585), ])  
Gheight<-max(Gband$Spec)  
  
#rescale, smooth to remove outliers and interporlate  
BR2nn<-as.data.frame(supsmu(BR2n$ramnum, BR2n$Spec, span = "0.03"))  
samplespectratable<-data.frame(ramnum = 1200:1799)  
InterpBR<-as.data.frame(approx(BR2nn[,1], BR2nn[,2], xout= samplespectratable$ramnum))  
InterpBR[is.na(InterpBR)]=min(InterpBR[,2], na.rm = TRUE)  
InterpBR$Spec<-rescale\_01(InterpBR$y)  
  
  
  
#Construct Look Up Table Number 1 (asphaltic Petroleum and Humic Acid)  
peat<-read.table(file ="peatextract.dat")  
silasph<-read.table(file ="siljianaspah.dat")  
  
#smooth and tidyspectra (use a smoothing function that will also interpolate)  
peat[,7:8]<-as.data.frame(supsmu(peat[,1], peat[,2], span = "0.03"))  
peat[,9]<-rescale\_01(peat[,8])  
silasph[,6:7]<-as.data.frame(supsmu(silasph[,1], silasph[,2], span = "0.02"))  
silasph[,8]<-rescale\_01(silasph[,7])  
  
  
#build table of end members  
spectratable<-data.frame(ramnum = 1008:1799)  
resampeat<-approx(peat[,1], peat[,9], xout= spectratable$ramnum)  
resamasph<-approx(silasph[,1], silasph[,8], xout= spectratable$ramnum)  
spectratable$peat<-resampeat$y  
spectratable$asph<-resamasph$y  
  
#calcuculate lookup table 1  
for (f in 1:100){  
 spectratable[,3+f]<-((f/100)\*spectratable$peat)+((100-f)/100)\*spectratable$asph  
}  
  
#subset the table for a quikcer/easier life  
#subset the spectratable from 1200 to 1799 (600 rows)  
spectratable<-subset(spectratable, ramnum > 1199 & ramnum < 1800)  
eval <-as.data.frame(c(1:(ncol(spectratable)-3)))  
  
  
#look up part fro asph/peat %  
#nn is the result for the amount of peat/humic acid or peat that best explains asphaltene   
for (j in 1:nrow(eval)){  
 eval[j,2] = cor(spectratable[,j+3],InterpBR$Spec,"complete.obs")  
}  
colnames(eval)<-c("percentPeat","Rcor")  
eval<-arrange(eval,(-Rcor),(percentPeat))  
eval[1,]

## percentPeat Rcor  
## 1 2 0.6611753

nn<-eval[1,1]  
  
  
  
#Construct Look Up Table Number 2 (asphaltic Petroleum and Humic Acid + pigment)  
#T4py is a mixture of of beta carotrene and c-phycocyanin spectra  
T4py<-read.table(file ="tetrapyrolemodel3.txt")  
  
#smooth and resample  
T4py[,3:4]<-as.data.frame(supsmu(T4py[,1], T4py[,2], span = "0.03"))  
T4py[,5]<-rescale\_01(T4py[,4])  
nspectratable<-data.frame(ramnum = 1200:1799)  
resamT4py<-approx(T4py[,1], T4py[,5], xout= spectratable$ramnum)  
  
  
  
procT4py<-data.frame(ramnum = 1008:1799)  
procT4py<-subset(procT4py, ramnum >= 1200)  
procT4py$T4pyspecl<-rescale\_01(resamT4py$y)  
  
  
#baseline removal for reference pigment - this can be skipped  
T4minl<-subset(procT4py, T4pyspecl == min(procT4py[1:30,2]))   
T4minh<-subset(procT4py, T4pyspecl == min(procT4py[(nrow(procT4py)-20):nrow(procT4py),2]))   
T4baslinepoints<-rbind(T4minl,T4minh)  
linearMod <- lm(T4pyspecl ~ ramnum, data=T4baslinepoints)  
baseline<-predict(linearMod, procT4py)  
procT4py[,3]<-baseline  
procT4py$T4pyspecl<-procT4py$T4pyspecl-procT4py$V3  
  
#tetrapryole mix  
for (g in 1:101){  
 nspectratable[,g+1]<-((g/100)\*spectratable[,nn+1])+((100-g)/100)\*procT4py$T4pyspecl  
}  
  
#look up part fro % bb  
neval <-as.data.frame(c(1:(ncol(nspectratable)-1)))  
for (i in 1:nrow(neval)){  
 neval[i,2] = cor(nspectratable[,i+1],InterpBR$Spec,"complete.obs")  
}  
colnames(neval)<-c("percentbb","Rcor")  
neval<-arrange(neval,(-Rcor),(percentbb))  
neval[1,]

## percentbb Rcor  
## 1 53 0.6893653

mn<-neval[1,1]  
  
#final fractions calculated  
BB<-1-(neval$percentbb[1]/100)  
Humic<-(1-BB)\*(eval$percentPeat[1]/100)  
Asph<-(1-BB)\*(1-(eval$percentPeat[1]/100))  
Chksum<-BB+Humic+Asph  
  
#gather results   
ReturnedData<-c(Asph, Humic, BB, Chksum, eval[1,2], neval[1,2],  
 Gheight, Gheight\*Asph)  
ReturnedData<-lapply(as.numeric(ReturnedData), round, digits = 2)  
ReturnedData<-c(paste(name\_of\_file),ReturnedData)  
results<-as.data.frame(rbind(ReturnedData))  
colnames(results)<-c("name", "Asphaltene","Humic Acid","pigments","Chksum","Asph. cor","Pig cor.",  
 "G-height","corrected G-height")  
results<-unlist(results)  
results<-as.data.frame(results)  
  
#write as a text file with the an altered file name  
write.table(results, file = paste("output",name\_of\_file), col.names = TRUE,  
 row.names = TRUE)