UV-Vis spectra synthesis - shiny app documentation

Nino Pozar

29 January 2019

Introduction

This is documentation supporting the **UV-Vis spectra synthesis app**. This app is primarily intended for analytical chemist for for usage in chemical analysis. The output of this analysis are the plot of UV-Vis spectra and the table of it's pKa values. The output of the app depends on two input parameters: number of pKa values and set seed value that ensures the reproducibility. With given input, the app constructs a spectra for 20 randomly chosen pH values and creates UV-Vis spectra as sum of Gaussian functions for given number of pKa values. The pKa values are randomly given in the interval 0-14.

Downloading and preparing the required packages

```
## Installing required packages
required_packages <- c("dplyr", "ggplot2", "tidyr", "knitr")
packages_to_install <- required_packages[!(required_packages %in% installed.packages()[,"Package"])]
if(length(packages_to_install)) install.packages(packages_to_install)
lapply(required_packages, require, character.only = TRUE)
```

Creating functions that will create the spectra

```
#Function creates concentration matrix for given arguments, also returns pK values and total concentrat
conc_matrix <- function(n_pk, pH, pK=runif(n_pk,0,14), ctot=1) {
  pK <- sort(pK) #pKa values must in ascending order
  K <- 10^-(pK)
  H <- 10^-pH
  cmat <- matrix(nrow=length(H),ncol=(n_pk+1))
  nazivnik <- vector("numeric", length(pH))
  for (i in 1:(n_pk+1)) {
    fac <- H^(i-1)
    if (i!=n_pk+1) {
      for (j in 1:(n_pk+1-i)) {
        fac <- fac*K[j]
      }
    }
    cmat[,(n_pk+2-i)] <- fac
    nazivnik <- nazivnik+fac
  }
  cmat <- cmat/nazivnik*ctot
  names(pK) <- paste("pKa",1:n_pk, sep="_")
  return(list(cmat=cmat, pK=pK, ctot=ctot))
}

#GaussF creates Gauss function with maximum peak (height), position (mu) and spread (sigma)
GaussF <- function(height,mu,sigma, start=200, finish=800) {
```

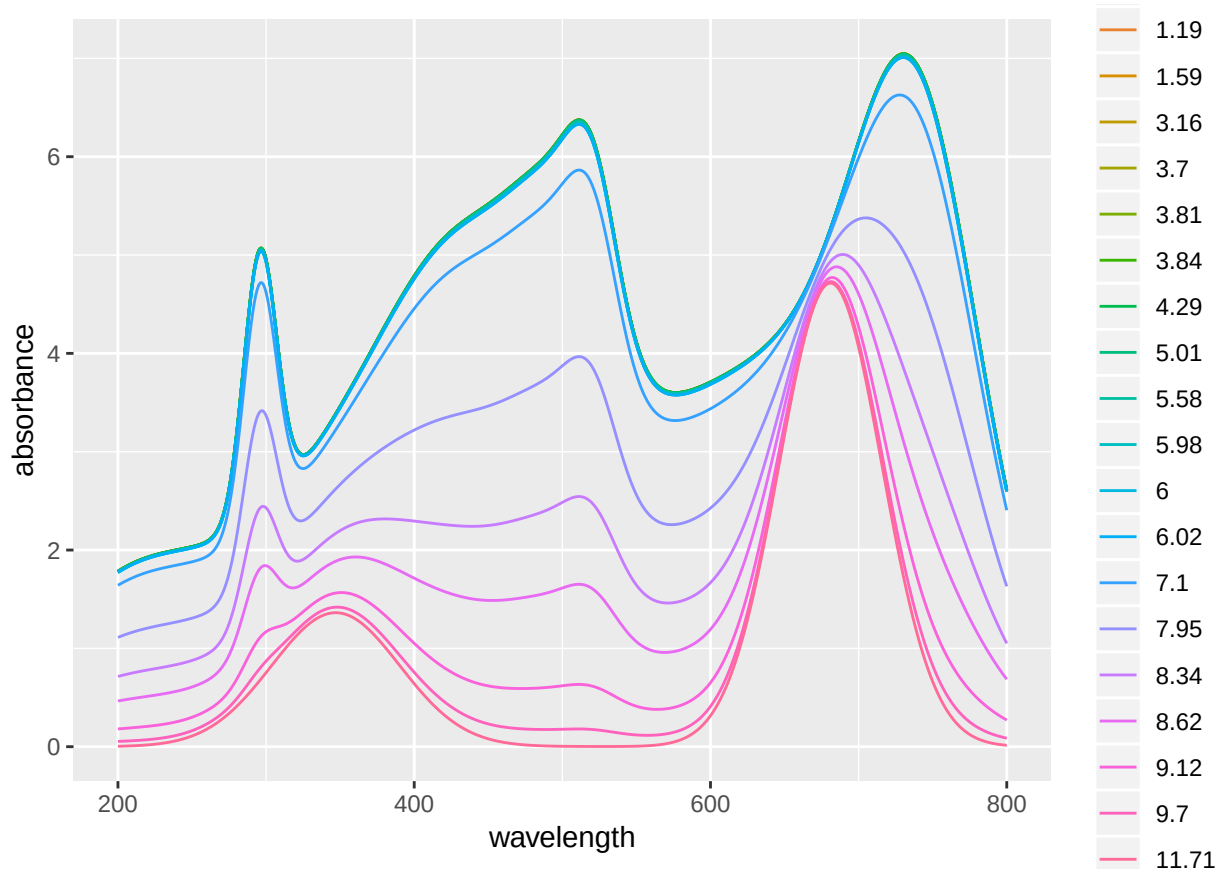
```
  x <- start:finish
  G <- height*exp(-1/2*((x-mu)/sigma)^2)
}

#G_spectra creates spectrum of pure specie, by summing Gaussian functions
#Arguments sd_sf and max are vectors of 2 elements- minimum and maximum for runif
#n_pik represents number of peaks in spectrum
G_spectra <- function(n_pik=sample(1:10,1), start=200, finish=800, sd_sf=c(10,80), max=c(0.5,5)) {
  spectra <- vector("numeric", (finish-start+1))
  for (i in 1:n_pik) {
    pik <- sample(start:finish,1)
    sd <- runif(1,sd_sf[1],sd_sf[2])
    height <- runif(1,max[1],max[2])
    spectra <- spectra+GaussF(height,pik,sd,start,finish)
  }
  return(spectra)
}

#Function creates matrix of pure species spectra for given number of pKa values
pure_spec <- function(n_pk, start=200, finish=800, ...) {
  n_pk <- n_pk+1
  all_spectra <- matrix(nrow=n_pk, ncol=(finish-start+1))
  for (i in 1:n_pk) {
    all_spectra [i,] <- G_spectra(start=start, finish=finish, ...)
  }
  colnames(all_spectra) <- paste(start:finish, " nm", sep="")
  rownames(all_spectra) <- paste("form",1:n_pk, sep="_")
  return(all_spectra)
}

#Function creates complete data of spectra for given pKas and pH
#Returns spectra, pH, pK and pure species spectra
spectras <- function(n_pk, pH, pK=runif(n_pk,0,14), ctot=1, ...) {
  pure_spectra <- pure_spec(n_pk, ...)
  c_list <- conc_matrix(n_pk,pH, pK, ctot)
  c_matrix <- c_list$cmat
  spectra <- c_matrix %*% pure_spectra
  colnames(spectra) <- colnames(pure_spectra)
  rownames(spectra) <- paste("sample", seq_along(pH), sep="_")
  names(pH) <- paste("sample", seq_along(pH), sep="_")
  return(list(spectra=spectra, pK=c_list$pK, pH=pH, pure_spectra=pure_spectra))
}
```

Example analysis for number of pKas=1 and set.seed=10

Here we do the example analysis of the app, for number of pKas=1 and set.seed=10. First, 20 pH values are simulated from the uniform distribution in range 0-14. Secondly spectras are created for 1 pKa value randomly chosen from uniform distribution in range 0-14. And finally in the end spectra are plotted using ggplot and table of simulated pKa values is given.

```
i <- 1
sseed <- 10
set.seed(sseed)
```

```
pH <- runif(20,0,14)
x <- spectras(i, pH, pK=runif(i,0,14))
s <- as.data.frame(x$spectra)
names(s) <- 200:800
m <- ncol(s)
s <- s %>%
  gather(wavelength,absorbance) %>%
  mutate(pH=as.factor(rep(round(pH,2),601))) %>%
  mutate(wavelength=as.numeric(wavelength))
g <- ggplot(s, aes(x=wavelength, y=absorbance, group=pH, color=pH))
g+geom_line()
```




```
f <- data.frame(pK=paste("pKa", 1:i, sep=""), value=round(x$pK,2))
kable(f)
```

	pK	value
pKa_1	pKa1	8.17

Conclusions

Analytical chemists can use this given data to test their own prediction algorithms to predict pKa values, which could in turn be deployed in drug development or biotech companies as part of quality control systems.