Supplementary Information for A Simple Algorithm for Despiking Raman Spectra

Darren Whitaker and Kevin Hayes

March 2018

‘‘‘

The code as set out below is an implementation of a simple algortihm for Despiking Raman spectra.

It consists of two functions and a number of sample operations.

Function to calculate modified Z Scores

```
ModifiedZscore = function(x) {
  m = median(x, na.rm=TRUE)
  M = mad(x, na.rm=TRUE)
  z = (x - m) / M
  z
}
```

Function to annihalate spikes at locations z

```
fixer = function(y, z, ma=5) {
  n = length(y)
  yout = y
  z[1] = z[n] = 1
  spikes = which(z==1)
  for(i in spikes) {
    w = seq(max(1,i-ma),min(n,i+ma))
    w = w[z[w] == 0]
    yout[i] = mean(y[w])
  }
  yout
}
```

Example Usage

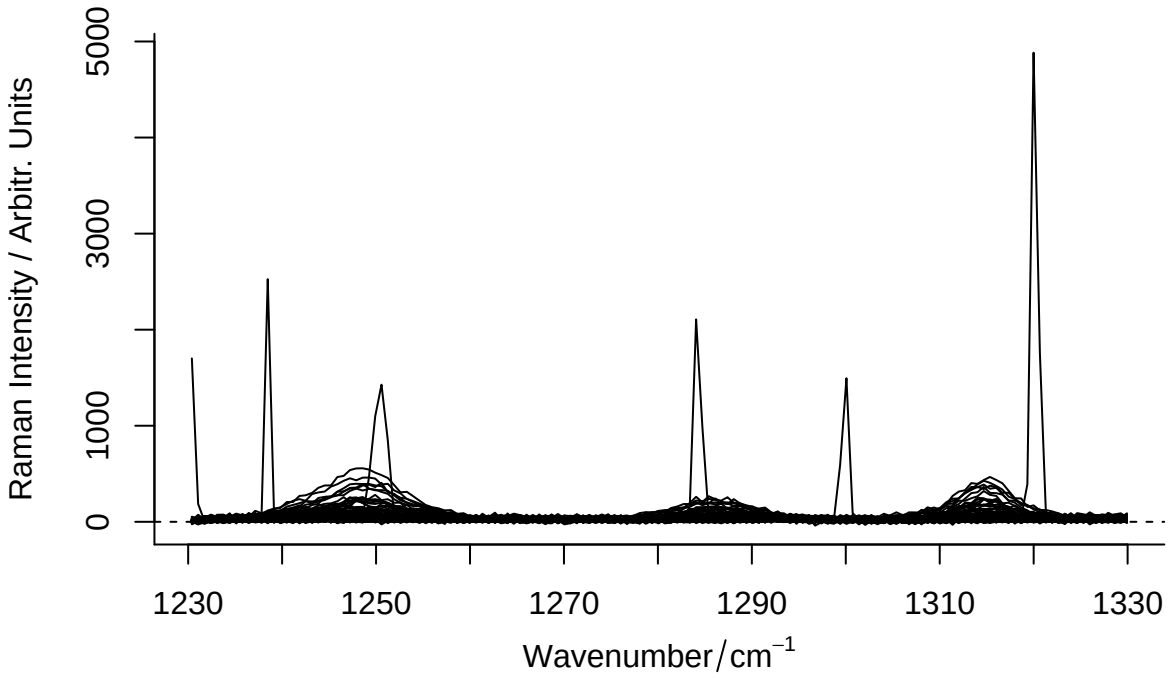
Spectral data are stored as a hyperSpec object for ease of subsequent analysis.

```
if (!require(hyperSpec)){
  install.packages(hyperSpec)
}

library(hyperSpec)
```

```
#setwd("\\path\\to\\directory") #This should be un-commented and used to set the working directory on t
load("TestData.RData")
```

Block 1: Example, blend 10 % API

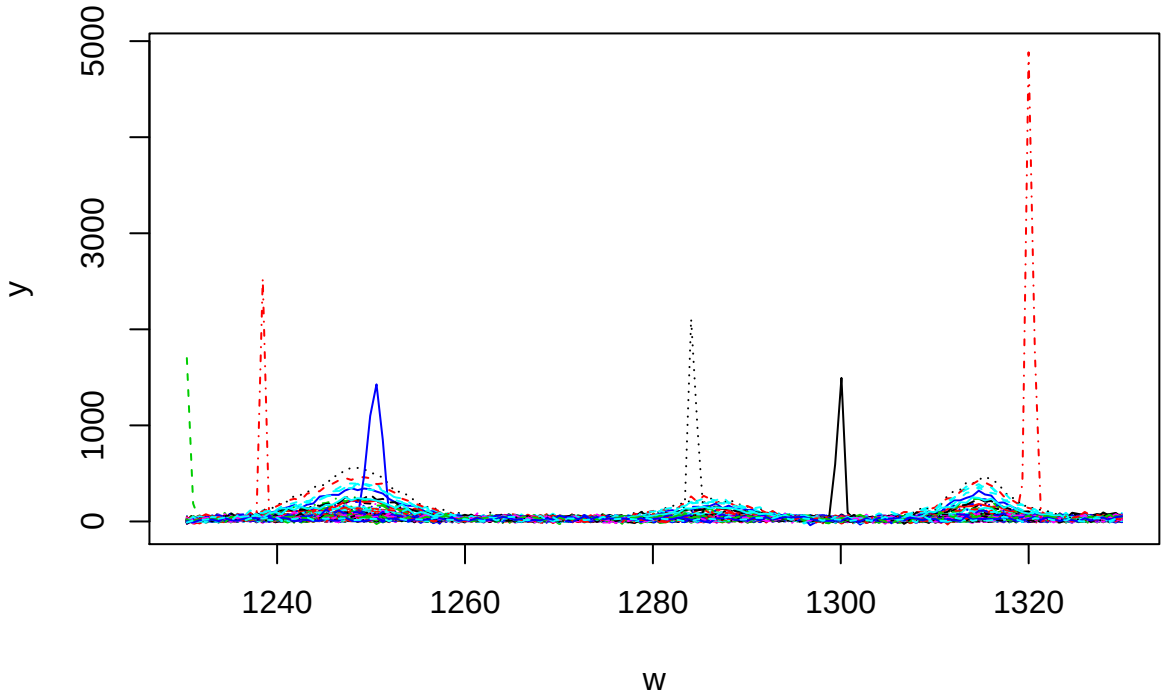
```
plot(test_data,spc.nmax=500)
```



Block 2: extract the data for blend 10 % API

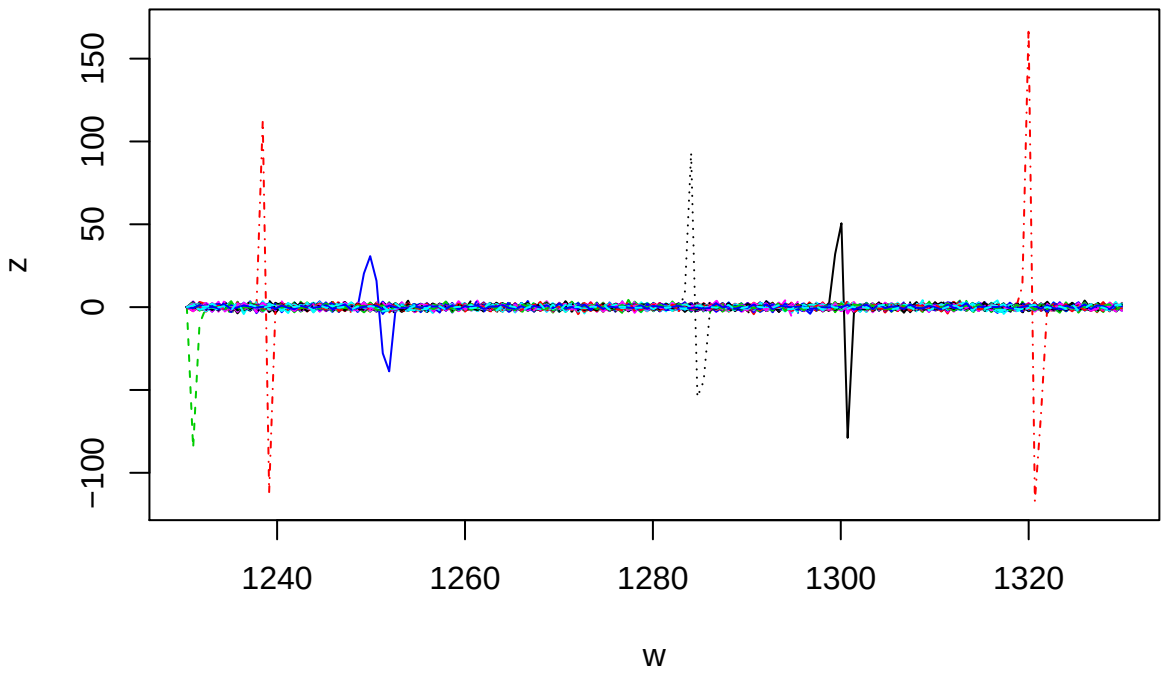
If data exists in hyperspec object the spectral matrix is extracted as below, otherwise the spectral matric can be passed directly

```
y = t(test_data[[]])
w = attr(test_data,"wavelength")

matplot(w,y,type="l")
```



Block 3: calculate modified Z Scores, and plot

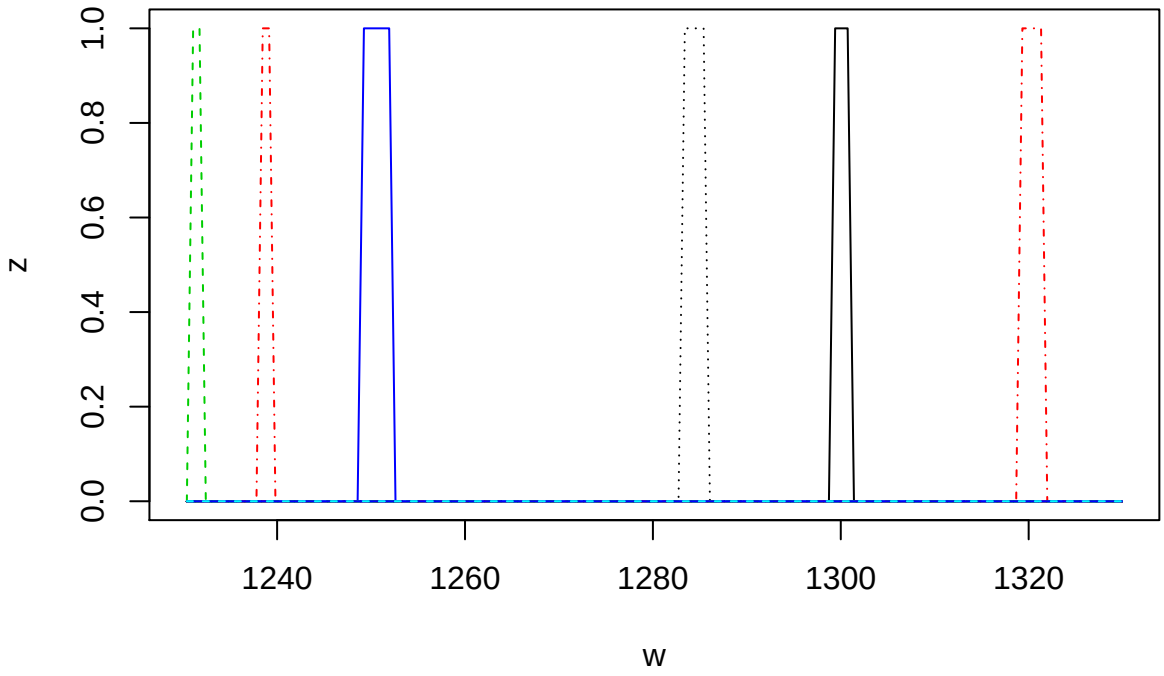
```
z = matrix(0, nrow(y)-1, ncol(y))

for(i in 1:ncol(y)) z[,i] = ModifiedZscore( diff(y[,i]) )

z = rbind(rep(0,ncol(y)),z)

matplot(w,z,type="l")
```



Block 4: create a matrix z identifying spikes

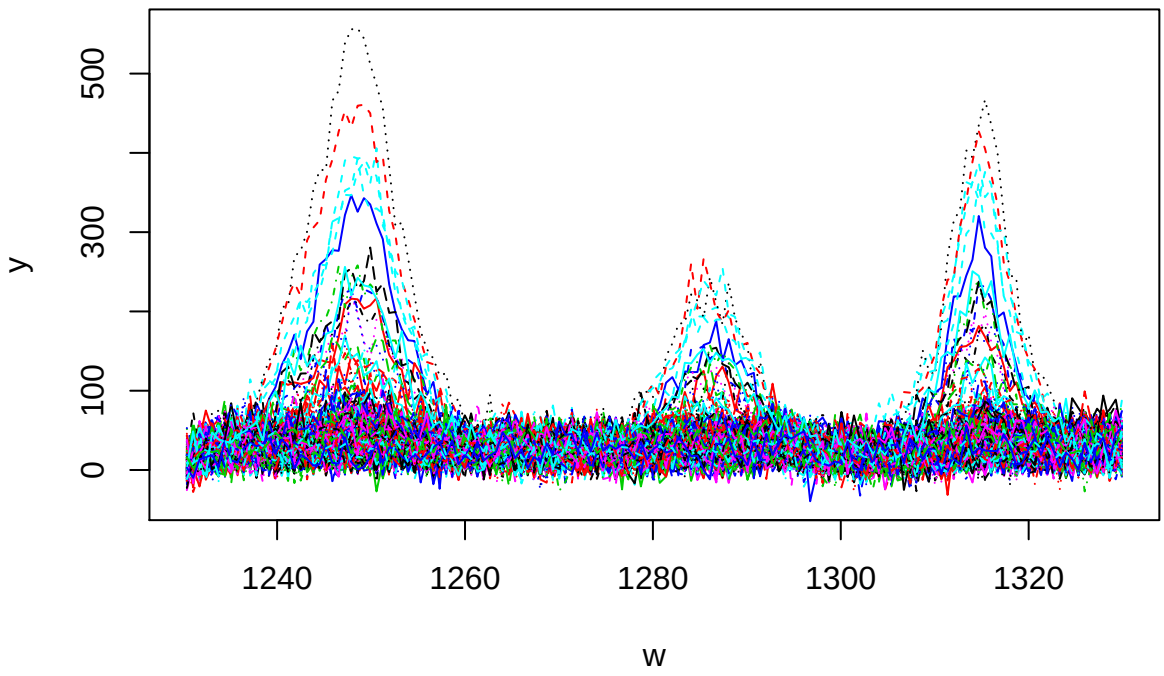
Here the user must choose a threshold, we recommend starting at a high value and decrease until optimal

```
threshold = 6

z = (abs(z) > threshold) * 1

matplot(w,z,type="l")
```



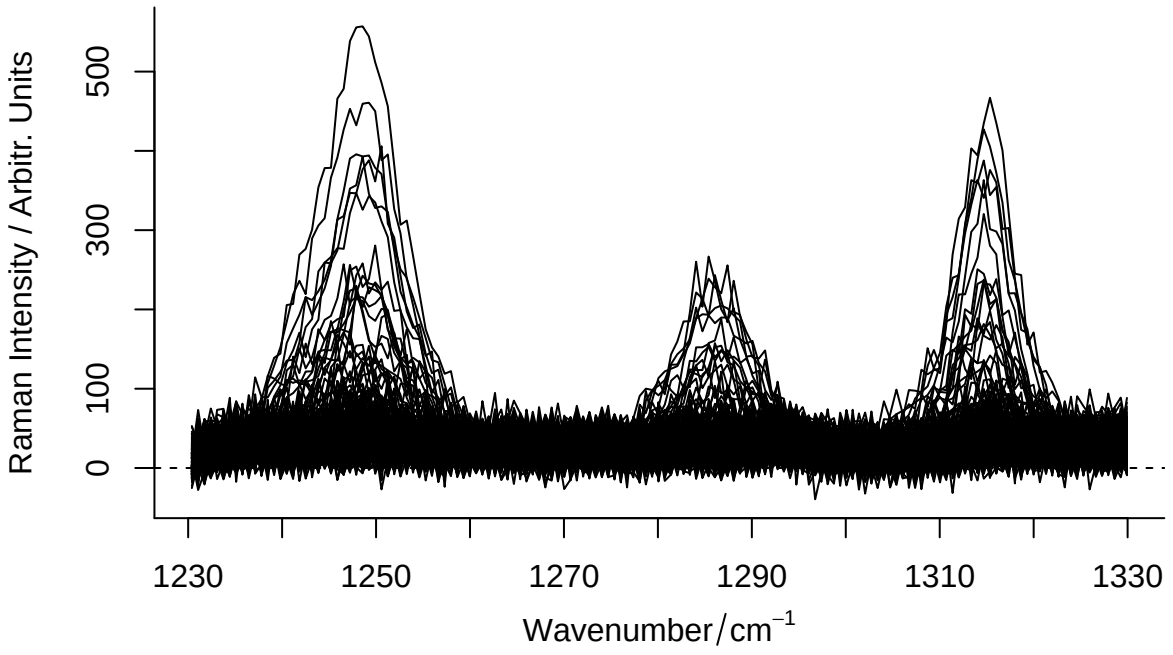
Block 5: despike the spiky spectra

```
spiky = seq(ncol(z)) [colSums(z) > 0]

for(i in spiky) y[,i] = fixer(y[,i], z[,i])

matplot(w,y,type="l")
```



Block 6: Assemble fixed hyperSpec object for easy plotting

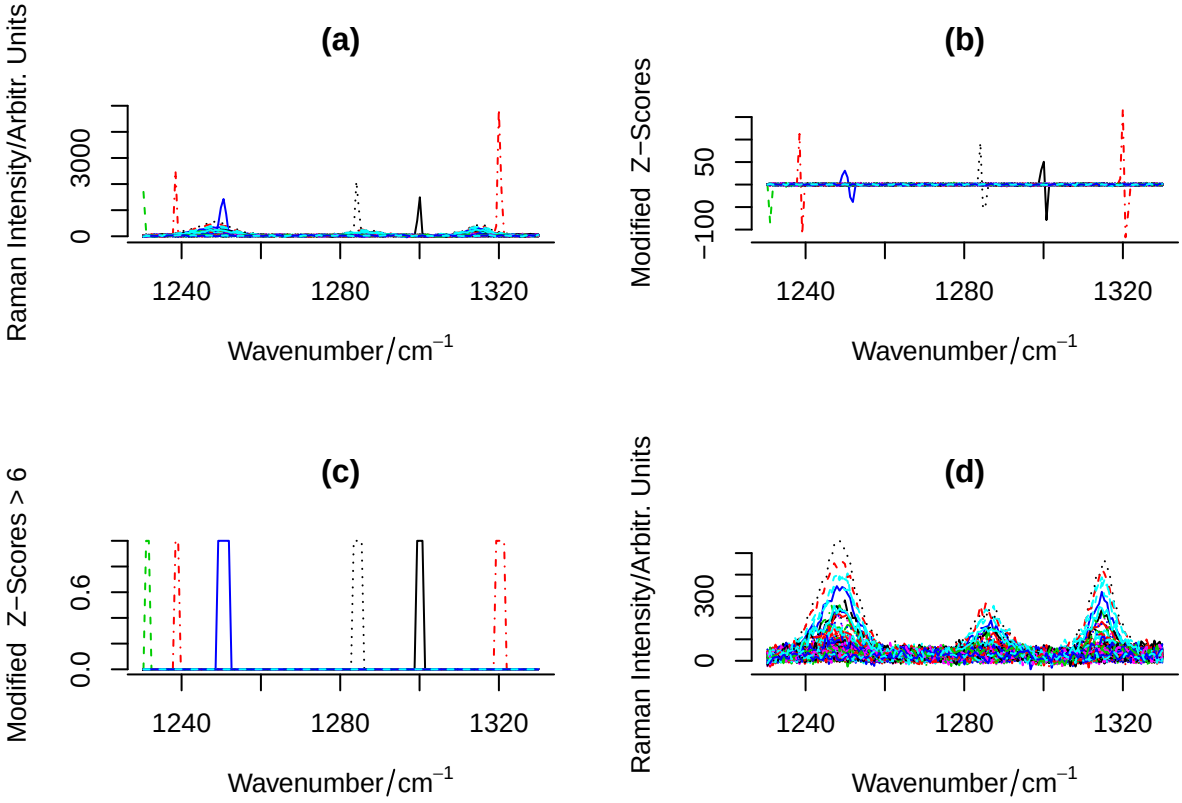
```
despiked_data <- test_data
despiked_data[[]] <- t(y)

plot(despiked_data, spc.nmax = 500)
```



Illustrative Figure from Paper

```
par(mfrow=c(2,2))

y = t(test_data[[]])
w = attr(test_data,"wavelength")

matplot(w,y,
        type="l",
        axes= FALSE,
        xlab= expression ("Wavenumber" / cm^-1),
        ylab="Raman Intensity/Arbitr. Units",
        main= "(a)")
axis(1,at= seq(from= 1220, to= 1340, by = 20))
axis(2)


z = matrix(0, nrow(y)-1, ncol(y))
for(i in 1:ncol(y)) z[,i] = ModifiedZscore( diff(y[,i]) )
z = rbind(rep(0,ncol(y)),z)
matplot(w,z,
        type="l",
        axes = FALSE,
        xlab= expression ("Wavenumber" / cm^-1),
```

```
        ylab= "Modified  Z-Scores",
        main= "(b)")
axis(1,at= seq(from= 1220, to= 1340, by = 20))
axis(2)


threshold = 6

z = (abs(z) > threshold) * 1

matplot(w,z,type="l",
        axes = FALSE,
        xlab= expression ("Wavenumber" / cm^-1),
        ylab= "Modified  Z-Scores > 6",
        main= "(c)")
axis(1,at= seq(from= 1220, to= 1340, by = 20))
axis(2)




spiky = seq(ncol(z)) [colSums(z) > 0]

for(i in spiky) y[,i] = fixer(y[,i], z[,i])

matplot(w,y,type="l",
        axes = FALSE,
        xlab= expression ("Wavenumber" / cm^-1),
        ylab="Raman Intensity/Arbitr. Units",
        main= "(d)")
axis(1,at= seq(from= 1220, to= 1340, by = 20))
axis(2)
```


(a)
Raman Intensity/Arbitr. Units
0
3000
1240
1280
1320
Wavenumber/cm^{-1}
(b)
Modified Z−Scores
−100
50
1240
1280
1320
Wavenumber/cm^{-1}
(c)
Modified Z−Scores > 6
0.0
0.6
1240
1280
1320
Wavenumber/cm^{-1}
(d)
Raman Intensity/Arbitr. Units
0
300
1240
1280
1320
Wavenumber/cm^{-1}