(Interactive) Spike Correction of Raman Spectra

Claudia Beleites <cbeleites@units.it>
CENMAT and DI3, University of Trieste

June 17, 2011

Contents

1 Introduction 1

2 Typical strategy for semi-automatic spike correction 1

1 Introduction

Raman spectra are often affected by cosmic rays hitting the CCD camera. These and similar sharp high intensity artifacts are called *spikes*.

- Spikes are often assumed to affect only one pixel
- this is not always the case: the camera at Uni. Trieste frequently produces wider spikes. (Possibly due to orientation in space?)

```
> library (hyperSpec)
> load ("cartilage-raw.RData")
```

2 Typical strategy for semi-automatic spike correction

spikefilter.R defines a semi-automatic spike finder:

```
> source ("spikefilter.R")
```

Function spikefilter filters each spectrum with c(-1, 2, -1), spikefilter2d applies the same filter also among spectra. This is useful if the spectra are rather similar: in that case, sharp bands are better distinguished from spikes that occur only in one spectrum.

A pre-processing that removes typical spectroscopic information may help. Particularly spikefilter2d benefits from making the spectra as similar as possible.

So I normalize the spectra to their median and then subtract the median spectrum:

```
> tmp <- sweep (cartilage, 1, median, `/`) # special abbreviation for hyperSpec's sweep:
>                                          # instead of a vector, we can give a function
> tmp <- sweep (tmp, 2, median , `-`)
```

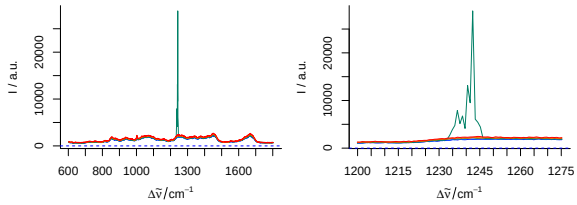


Figure 1: Spike in Raman spectrum: the green spectrum is affected by a spike. right: detail

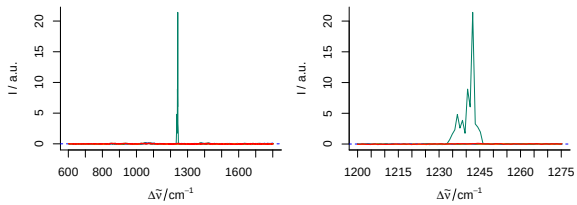


Figure 2: Pre-processing to enhance the spikes

Now the SNR with respect to the spikes is already greatly enhanced (2).

now generate spikiness scores:

```
> system.time (
+               scores <- spikefilter2d (spcmatrix= tmp [[]])
+               )

       User      System verstrichen
     10.280       1.360      11.829
```

now the interactive filtering, I take just the 1st 100 spectra to see the principle:

```
> spikes <- spikes.interactive (cartilage [1:100], scores [1:100, ])
> spikes <- which (is.na (spikes), arr.ind = TRUE) # spike points are set to NA
```

Note that suspicion no. 17 (spectrum 1) is a real sharp signal of an inorganic salt (carbonate), no spike.

After about 25 suspicions, the procedure can be ended: no more spikes but only real signals are found.

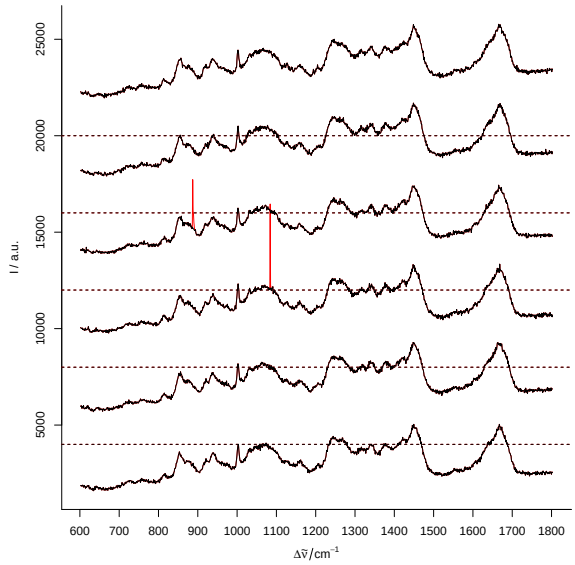
These are the points I considered spikes:

```
> spikes <- structure(c(67L, 67L, 67L, 67L, 67L, 67L, 67L, 67L, 67L, 67L,
+ 67L, 36L, 36L, 36L, 13L, 13L, 13L, 13L, 13L, 75L, 62L, 62L, 62L,
+ 62L, 12L, 12L, 86L, 86L, 86L, 86L, 86L, 86L, 43L, 43L, 43L, 43L,
+ 7L, 8L, 9L, 10L, 11L, 12L, 13L, 14L, 22L, 23L, 24L, 99L, 100L,
+ 101L, 281L, 282L, 283L, 284L, 285L, 350L, 421L, 422L, 423L, 424L,
+ 482L, 483L, 532L, 533L, 534L, 535L, 536L, 537L, 692L, 693L, 694L,
+ 695L), .Dim = c(36L, 2L), .Dimnames = list(NULL, c("row", "col"
+ )))
```

The results can be transferred to *cartilage* by

```
> tmp <- cartilage
> cartilage [[spikes, wl.index = TRUE]] <- NA
```

(the spikes matrix handed back by spikes.interactive could be assigned directly: cartilage [[1 : 100]] <- spikes, but that would produce an extremely large dput chunk here.)


I / a.u.
5000
10000
15000
20000
25000
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
Δṽ/cm⁻¹