Raman Spectra of Chondrocytes in Cartilage: hyperSpec's `chondro` data set

Claudia Beleites (cbeleites@units.it)
CENMAT, DMRN, University of Trieste

July 20, 2009

1 Introduction

This vignette describes the `chondro` data set. It shows a complete data analysis work flow on a Raman map demonstrating frequently needed preprocessing methods

- baseline correction
- normalization
- smoothing / interpolating spectra
- plotting spectra
- plotting false color maps

and other basic work techniques

- cutting the spectral range,
- selecting (extracting) or deleting spectra, and
- *aggregating* spectra (e.g. calculating cluster mean spectra).

The chemometric methods used are

- Principal Component Analysis (PCA) and
- hierarchical cluster analysis,

showing how to use data analysis procedures provided by R and other packages.

2 The Data Set

Raman spectra of a cartilage section were measured on each point of a grid, resulting in a so-called *Raman map*. Figure 1 shows a microscope picture fo the measuarea and its surroundings.

The measurement parameters were:

Excitation wavelength: 633 nm

Exposure time: 10 s per spectrum

Objective: 100×, NA 0.85

Measurement grid: 35 ×21 µm, 1 µm step size

Spectrometer: Renishaw InVia

All data to reproduce this Vignette is accessible at *hyperSpec*'s homepage, http://r-forge.r-project.org/projects/hyperspec/, as the original file is far too large to be included in the package.

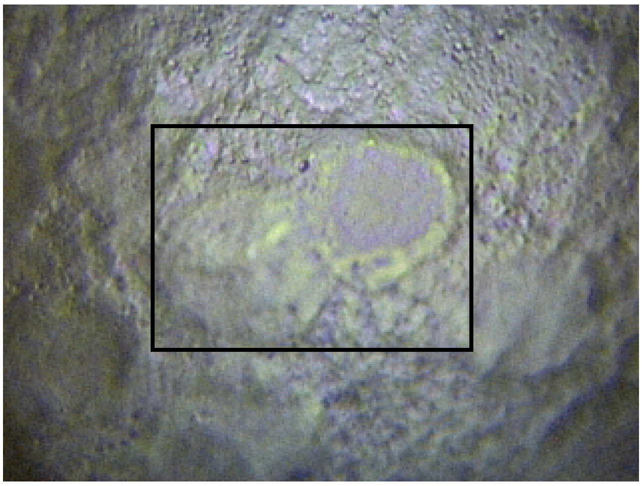


Figure 1: Microscope view of the cartilage section. The frame indicates the measurement area (35 ×21 µm).

3 Data Import

Renishaw provides a converter to export their proprietary data in a so-called long format ASCII file. Raman maps are exported having four columns, *y*, *x*, *raman shift*, and *intensity*. *hyperSpec* comes with a function to import such files, `scan.txt.Renishaw`. The function assumes a map as default, but can also handle single spectra (`data = "spc"`), time series (`data = "ts"`), and depth profiles (data = "depth"). In addition, large files may be processed in chunks. In order to speed up the reading `scan.txt.Renishaw` does not allow missing values, but it does work with `NA`.

```
> chondro <- scan.txt.Renishaw ("chondro.txt", data = "xyspc")
> chondro

hyperSpec object
   875 spectra
   3 data columns
   1272 data points / spectrum
wavelength: tilde(nu)/cm^-1 [numeric 1272]  601.622 602.664 ... 1802.15
data:  (875 rows x 3 columns)
   (1) y: y/(mu * m) [numeric 875] range -4.77 -3.77 ... 19.23
   (2) x: x/(mu * m) [numeric 875] range -11.55 -10.55 ... 22.45
   (3) spc: I / a.u. [AsIs matrix 875 x 1272] range 52.2573 52.5012 ... 1884.25 + NA
```

To get an overview of the spectra, :

```
> plot (chondro, "spcprctl5")
```

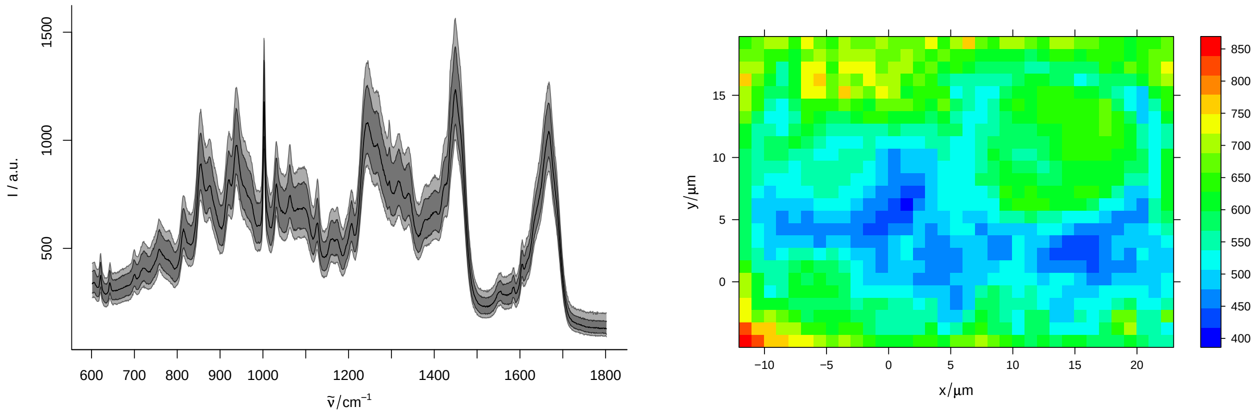
A sum intensity map is produced by:

```
> print (plotmap (chondro, na.rm = TRUE))
```

Figure 2 shows the results.

4 Preprocessing

As usual in Raman spectroscopy of biological tissues, the spectra need some preprocessing.



(a) The raw spectra: median, 16th and 84th, and 5th and 95th percentile spectra.

(b) The sum intensity of the raw spectra.

Figure 2: The raw data.

4.1 Spectral Smoothing

As the overview print shows that the spectra contain NAs (from cosmic spike removal that was done previously), the first step is to remove these. Another issue that can be solved at the same time is that the wavelength axis is not evenly spaced (the data points are between 0.85 and 1 cm^{-1} apart from each other). Furthermore, it would be good to trade some spectral resolution for hgher signal to noise ratio. All three of these issues are tackled by interpolating and smoothing of the wavelength axis by `spc.loess`. The resolution is to be reduced to 8 cm^{-1}, or 4 cm^{-1} data point spacing.

```
> chondro <- spc.loess (chondro, seq (602, 1800, 4))
> chondro
```

```
hyperSpec object
   875 spectra
   3 data columns
   300 data points / spectrum
wavelength: tilde(nu)/cm^-1 [numeric 300]  602 606 ... 1798
data:  (875 rows x 3 columns)
   (1) y: y/(mu * m) [numeric 875] range -4.77 -3.77 ... 19.23
   (2) x: x/(mu * m) [numeric 875] range -11.55 -10.55 ... 22.45
   (3) spc: I / a.u. [matrix 875 x 300] range 80.04420 81.75761 ... 1858.881
```

4.2 Baseline Correction

Next, we do a linear baseline correction. `spc.fit.poly.below` tries to automatically find appropriate support points for polynomial baselines. The default is a linear baseline, which is appropriate in our case:

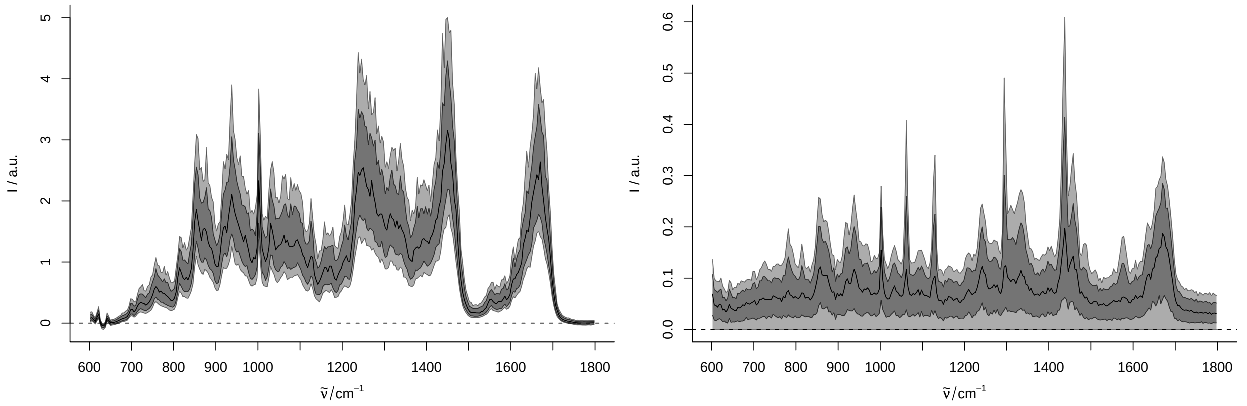
```
> baselines <- spc.fit.poly.below (chondro)
```

```
Fitting with npts.min =  15
```

```
> chondro <- chondro - baselines
```

4.3 Normalization

As the spectra are quite similar, area normalization should work well:.



(a) The spectra after smoothing, baseline correction, and normalization.

(b) The spectra after substracting the 5th percentile spectrum.

Figure 3: The preprocessed spectra.

```
> chondro <- sweep (chondro, 1, apply (chondro, 1, mean), "/")
> plot (chondro, "spcprctl5")
```

For the results of these preprocessing steps, see figure 3a.

4.4 Substracting the Overall Composition

The spectra are very homogeneous, but I'm interested in the differences between the different regions of the sample. Sustracting the minimum spectrum cancels out the matrix compositon that is common to all spectra. But the minimum spectrum also picks up a lot of noise. So instead, the 5th percentile spectrum is substracted:

```
> chondro <- sweep (chondro, 2, apply (chondro, 2, quantile, 0.05), "-")
> plot (chondro, "spcprctl5")
```

The resulting data set is shown in figure 3b. Some interesting differences start to show up: there are distinct lipid bands in some but not all of the spectra.

4.5 Outlier Removal by Principal Component Analysis (PCA)

PCA is a technique that decomposes the data into scores and loadings (virual spectra). It is known to be quite sensitive to outliers. Thus, I use it for outlier detection. The resulting scores and loadings are put again into *hyperSpec* objects by `decomposition`:

```
> pca <- prcomp (~ spc, data = chondro$., center = TRUE)
> scores <- decomposition (chondro, pca$x, label.wavelength = "PC", label.spc = "score / a.u.")
> loadings <- decomposition (chondro, t(pca$rotation), scores = FALSE, label.spc = "loading I / a.u.")
```

Plotting the scores of each PC against all other gives a good idea where to look for outliers.

```
> pairs (scores [[,,1:20]], pch = 19, cex = 0.5)
```

Now the spectra can be found either by plotting two scores against each other (by `plot`) and identifying with `identify`, or they can be identified in the score map by `map.identify`. There is also a function to identify spectra in a spectra plot, `spc.identify`, but this is not helpful here.

```
> out <- map.identify (scores [,,5])
> out <- c (out, map.identify (scores [,,6]))
> out <- c (out, map.identify (scores [,,7]))
```

```
> out
```

```
[1] 105 140 216 289  75  69
```

```
> outcols <- c ("red", "blue", "#800080", "orange", "magenta", "brown")
> cols <- rep ("black", nrow(chondro))
> cols [out] <- outcols
```

We can check our findings by comparing the spectra to the bulk of spectra (figure):

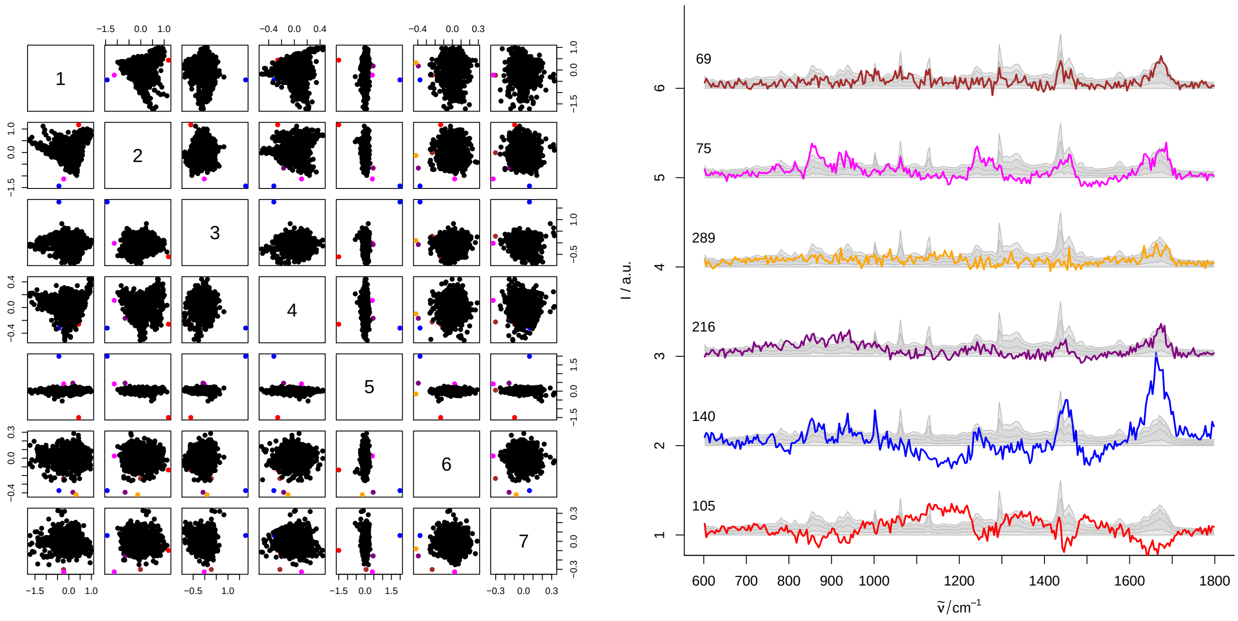
```
> plot(chondro[1], plot.args = list (ylim = c (1, length (out) + .7)), lines.args = list(  type = "n"))
> for (i in seq (along = out)){
+    plot(chondro, "spcprctl5", yoffset = i, add = TRUE, col = "gray")
+    plot (chondro [out[i]], yoffset = i, col = outcols[i] , add = TRUE, lines.args = list (lwd = 2))
+    text (600, i + .33, out [i])  }
```

and also by looking where these spectra appear in the scores pairs plot (figure):

```
> pairs (scores [[,,1:7]], pch = 19, cex = 1, col = cols)
```

Finally, the outliers are removed:

```
> chondro <- chondro [- out]
```



(a) pairs plot of the first 7 scores.

(b) The suspected outlier spectra.

Figure 4: Outlier removal by PCA

5 Hierarchical Cluster Analysis (HCA)

HCA fuses objects according to their (dis)similarity. The result is a dendrogram, a graph stating at which level two objects are similar and thus grouped together.

The first step in HCA is the choice of the distance. The R function `dist` offers a variety of distance measures to be computed. The so-called PEARSON distance $D^2{}_{Pearson} = \frac{1-COR(X)}{2}$ is popular in data analysis of vibrational spectra and is provided by *hyperSpec*.

Also for computing the dendrogram, a number of choices are available. I choose WARD's method, and, as it uses EUCLIDean distance for calculating the dendrogram, EUCLIDean distance also for the distance matrix :

```
> dist <- dist (chondro [[]])
> dendrogram <- hclust (dist, method = "ward")
```

```
> plot (dendrogram)
```

In order to get clusters, the dendrogram is cut at a level specified either by height or by the number of clusters. The result for k= 3 clusters is plot as a map. If `plotmap`'s z is a factor, the legend bar does not show intermediate colors.

```
> clusters <- cutree (dendrogram, k = 3)
> print (plotmap (chondro, z = as.factor (clusters)))
```

The cluster membership can also be marked in the dendrogram:

```
> plot (dendrogram, labels = FALSE, hang = 0)
> col.clust <- matlab.palette(3)
> points (seq_along (dendrogram$order), rep (-3, length (dendrogram$order)),
+         col = col.clust [clusters [dendrogram$order]], pch = "|")
```

Figure 5a shows the dendrogram and 5b the resulting cluster map. The three clusters correspond to the cartilage matrix, the lacuna and the cells. The left cell is destroyed and its contents are leaking into the matrix, while the right cells looks intact.

We can calculate the cluster mean spectra using `aggregate`. However, we can do even better and plot the cluster mean spectra ± 1 standard deviation (see figure 6a):

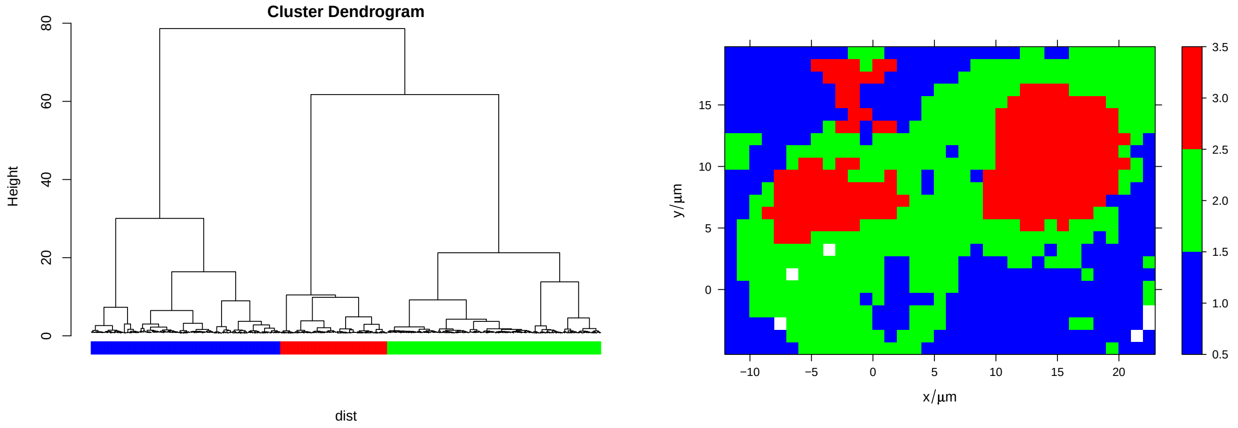
```
> cluster.means <- aggregate (chondro, clusters, mean_pm_sd)
> plot(cluster.means, yoffset = rep ((1:3), each = 3), col = rep (matlab.palette (3), each = 3))
```

6 Plotting a False-Colour Map of Certain Spectral Regions

hyperSpec comes with a sophisticated inferface for specifying spectral ranges. Expressing things like 1000 cm^{-1} ± 1 data points are easily possible. Thus, we can have a fast look at the nucleic acid distribution, using the DNA bands at 728, 782, 1098, 1240, 1482, and 1577 cm^{-1}:

```
> print (plotmap (chondro[, , c( 728 - 1i ~  728 + 1i,
+                                782 - 1i ~  782 + 1i,
+                               1098 - 1i ~ 1098 + 1i,
+                               1240 - 1i ~ 1240 + 1i,
+                               1482 - 1i ~ 1482 + 1i,
+                               1577 - 1i ~ 1577 + 1i)]))
```

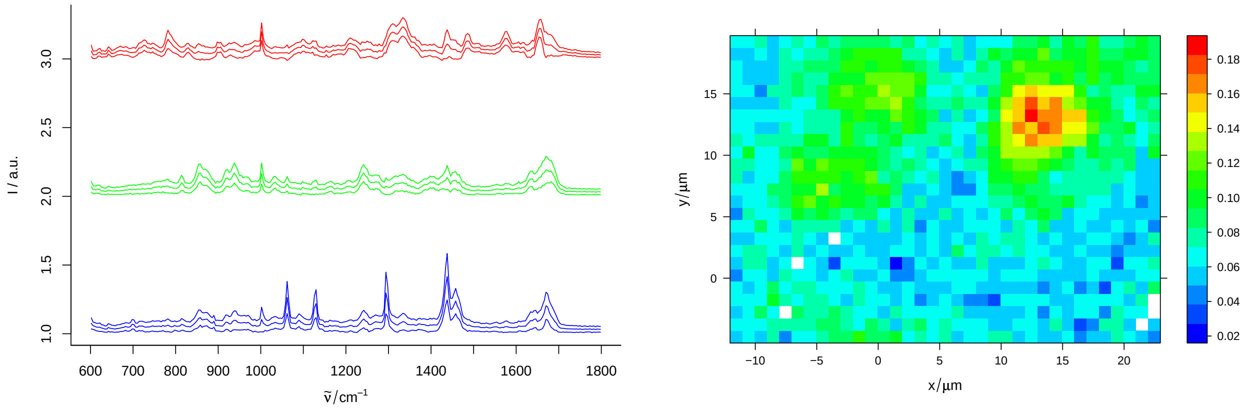
The result is shown in figure 6b. While the nucleus of the right cell shows up nicely, nothing is detected in the remainders of the left cell.



(a) The dendrogram.

(b) The cluster map for $k = 3$ clusters.

Figure 5: Hierarchical cluster analysis.



(a) The cluster mean ± 1 standard deviation spectra. The blue cluster shows distinct lipid bnds, the green cluster collagen, and the red cluster proteins and nucleic acids.

(b) False colour map of the DNA band intensities.

Figure 6