Unstable Laser Emission
Vignette for the Data Set `laser` of the R package hyperSpec

Claudia Beleites <chemometrie@beleites.de>
CENMAT and DI3, University of Trieste
Spectroscopy · Imaging, IPHT Jena e.V.

February 20, 2014

Reproducing the Examples in this Vignette

All spectra used in this manual are installed automatically with *hyperSpec*.
The source data files can be found with the command:
`> system.file ("doc/src/rawdata/laser.txt.gz", package = "hyperSpec").`
In order to reproduce the examples by typing in the commands, have a look at the definitions in `vignettes.defs`.

Contents

1 Introduction 1
2 Loading the Data and Preprocessing 2
3 Inspecting the time dependency of the laser emission 3
4 False-colour plot of the spectral intensity over wavelength and time 4
5 3d plot of the spectral intensity over wavelength and time 4

Suggested Packages

rgl: available

1 Introduction

This data set consists of a time series of 84 spectra of an unstable laser emission at 405 nm recorded during ca. 1.5 h.

The spectra were recorded during the installation of the 405 nm laser at a Raman spectrometer. There is no Raman scattering involved in this data, but the Raman software recorded the abscissa of the spectra as Raman shift in wavenumbers.

This document shows

- How to convert the wavelength axis (spectral abscissa)
- How to display time series data as intensity over time diagram
- How to display time series data as 3d and false colour image of intensity as function of time and wavelength.

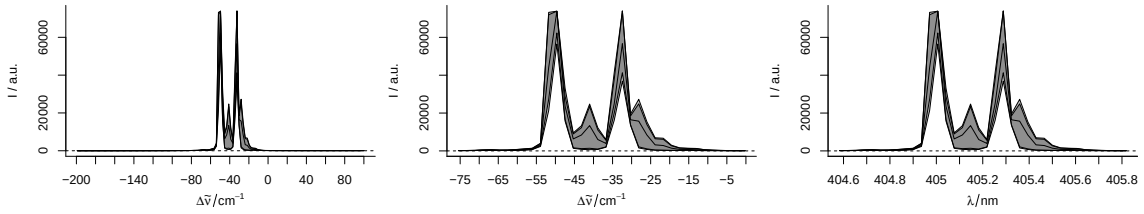
2 Loading the Data and Preprocessing

```
> laser <- scan.txt.Renishaw ("rawdata/laser.txt.gz", data = "ts")
> plot (laser, "spcprctl5")
```

As the laser emission was recorded with a Raman spectrometer, the wavelength axis initially is the Raman shift in wavenumbers (cm^{-1}).

As most of the spectra do not show any signal (fig. 1a), so the spectral range can be cut to -75 – 0 cm^{-1}. Note that negative numbers in the spectral range specification with the tilde do not exclude the spectral range but rather mean negative values of the wavelength axis. The results are shown in figure 1b.

```
> laser <- laser [,,-75~0]
> plot (laser, "spcprctl5")
```



(a) The raw spectra. (b) The cut spectra. (c) The spectra with wavelength axis.

Figure 1: The laser emission spectra.

The wavelength axis was recorded as Raman shift from 405 nm. However, the spectra were taken before calibrating the wavelength axis. The band at -50 cm^{-1} is known to be at 405 nm.

```
> wl (laser) <- wl (laser) + 50
```

Furthermore, as the spectra are not Raman shift but emission, the wavelength axis should be converted to proper wavelengths in nm.

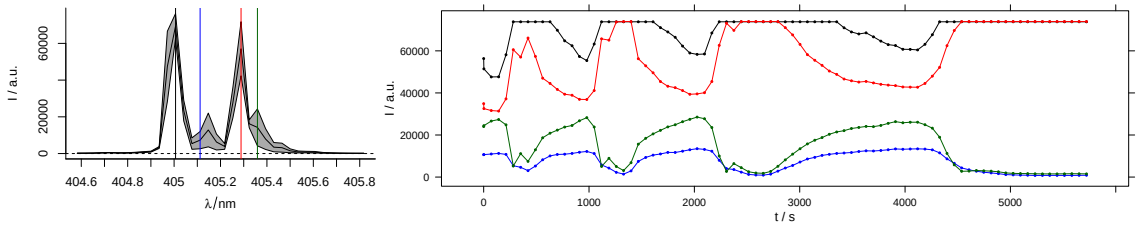
The Raman shift is calculated from the wavelength as follows:

$$\Delta\tilde{\nu} = \frac{1}{\lambda_0} - \frac{1}{\lambda}$$

with $\Delta\tilde{\nu}$ being the Raman shift, and λ_0 the excitation wavelength for a Raman process, here 405 nm.

The wavelengths corresponding to the wavenumbers are thus:

$$\lambda = \frac{1}{\frac{1}{\lambda_0} - \Delta\tilde{\nu}}$$



(a) (b)

Figure 2: The laser emission time series. 2a shows the spectral position of the bands. The time series are plotted in corresponding colors in 2b.

Taking into account that 1 cm = 10^7 nm, we arrive at the new wavelength axis:

```
> wl (laser) <- list (
+     wl = 1e7 / (1/405e-7 - wl (laser)),
+     label = expression (lambda / nm)
+     )
```

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

Note that the new wavelength axis label is immediately assigned as well.

Now, save `laser` as the `laser` data set shipped with *hyperSpec*.

```
> save (laser, file = "laser.rda")
> laser
```

```
hyperSpec object
   84 spectra
   2 data columns
   36 data points / spectrum
wavelength: lambda/nm [numeric] 404.58 404.62 ... 405.82
data:  (84 rows x 2 columns)
   1. t: t / s [numeric] 0 2 ... 5722
   2. spc: I / a.u. [matrix36] 164.65 179.72 ... 112.09
```

3 Inspecting the time dependency of the laser emission

The maxima of the different emission lines encountered during this measurement are at 405.0, 405.1, 405.3, and 405.4 nm (fig. 2a).

Alternatively they can be extracted from the graph using `locator` which reads out the coordinates of the points the user clicks with the mouse (use middle or right click to end the input):

```
> wls <- locator()$x
```

```
> plot (laser, "spcmeansd")
> cols <- c("black", "blue", "red", "darkgreen")
> abline (v = wls, col = cols )
```

`plotc` can also be used to plot time-series. In that case, the abscissa needs to be specified in parameter `use.c`. The collection time is stored in column `$t` in seconds from start of the measurement, and can be handed over as the column name. The resulting time series are shown in figure 2b variable,

```
> plotc (laser [,, wls], spc ~ t, groups = .wavelength, type = "b",  cex = 0.3, col = cols)
```

Another option is to condition the plot on λ:

```
> plotc (laser [,, wls], spc ~ t | .wavelength, type = "b", cex = 0.3, col = "black")
```

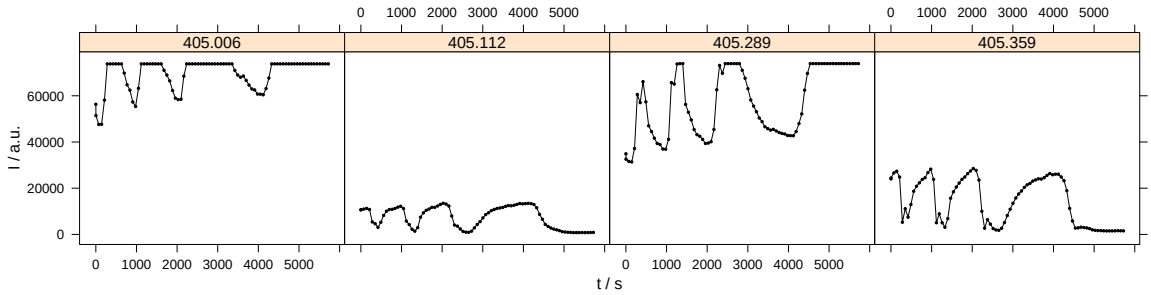


Figure 3: The time series plots can also be conditioned on `$.wavelength`.

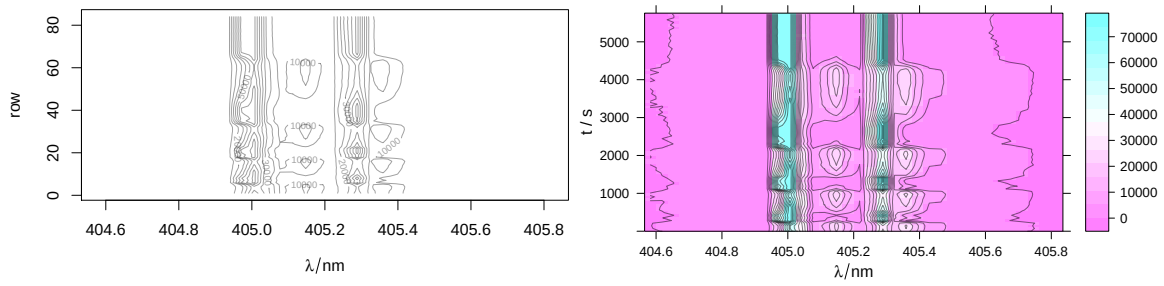
4 False-colour plot of the spectral intensity over wavelength and time

hyperSpec supplies functions to draw the spectral matrix using *lattice*'s `levelplot`.

```
> plot (laser, "mat", contour = TRUE, col = "#00000060")
```

hyperSpec's `levelplot` method can be used to display the spectra matrix over a data column (instead of the row number): figure 4. Note that the *hyperSpec* object is the *second* argument to the function (according to the notation in `levelplot`).

```
> levelplot (spc ~ .wavelength * t, laser, contour = TRUE, col ="#00000080")
```



(a) (b)

Figure 4: The spectra matrix of the `laser` data set. The ordinate of the plot may be the number of the spectrum accessed by `$.row` (a) or any other extra data column, here `$t` (b).

5 3d plot of the spectral intensity over wavelength and time

hyperSpec objects can easily be drawn with *rgl*[1]:

```
> require (rgl)
```

rgl's `persp3d` plots a surface in 3d defined by points in x, y, and z. Handing over the appropriate data columns of the *hyperSpec* object is easy (fig. 5):

```
> laser <- laser [,,404.8 ~ 405.6] / 10000
> laser$t <- laser$t / 3600
> cols <- rep (matlab.palette (nrow (laser)), nwl (laser))
> surface3d(y = wl(laser), x = laser$t, z = laser$spc, col =  cols)
> surface3d(y = wl(laser), x = laser$t, z = laser$spc + .1 * min (laser),
```

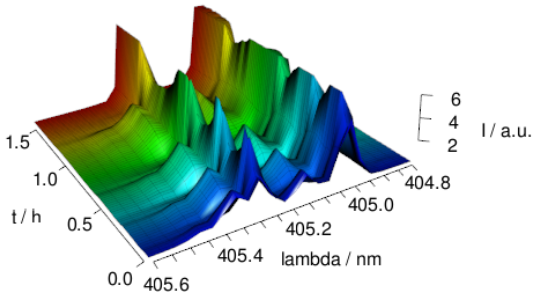


Figure 5: The 3d plot of the laser data

```
+           col =  "black", alpha = .2,front = "lines", line_antialias = TRUE)
> aspect3d (c(1, 1, 0.25))
> axes3d (c ('x+-', 'y--', 'z--'))
> axes3d ('y--', nticks = 25, labels= FALSE)
> mtext3d ("t / h",     'x+-', line = 2.5)
> mtext3d ("lambda / nm",   'y--', line = 2.5)
> mtext3d ("I / a.u.", 'z--', line = 2.5)
```

References

[1] Daniel Adler, Duncan Murdoch, and others. *rgl: 3D visualization device system (OpenGL)*, 2014. URL http://CRAN.R-project.org/package=rgl. R package version 0.93.996.

Session Info

```
R version 3.0.2 Patched (2014-02-18 r65029)
Platform: x86_64-unknown-linux-gnu (64-bit)

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C               LC_TIME=en_US.UTF-8
 [4] LC_COLLATE=C               LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C                  LC_ADDRESS=C
[10] LC_TELEPHONE=C             LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] tools     grid      stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] rgl_0.93.996            ggplot2_0.9.3.1         baseline_1.1-2          plotrix_3.5-3
[5] MASS_7.3-29             hyperSpec_0.98-20140219 mvtnorm_0.9-9997        lattice_0.20-24

loaded via a namespace (and not attached):
 [1] RColorBrewer_1.0-5 SparseM_1.03       colorspace_1.2-4   dichromat_2.0-0    digest_0.6.4
 [6] gtable_0.1.2       labeling_0.2       munsell_0.4.2      plyr_1.8           proto_0.3-10
[11] reshape2_1.2.2     scales_0.2.3       stringr_0.6.2
```