Fitting Polynomial Baselines to Complete Spectra

C. Beleites, DMRN, Università degli Studi di Trieste, Trieste/I

July 28, 2009

Syntax & parameters

`spc.fit.poly.below (fit.to, apply.to = fit.to, poly.order = 1, npts.min = NULL, noise = 0, short = NULL, user = NULL, date = NULL)`

`fit.to`: `hyperSpec` object with the spectra whose baselines are to be fitted.

`apply.to`: `hyperSpec` object giving the spectral range, on wich the baselines should be evaluated.
If `apply` is `NULL`, a `hyperSpec` object with the polynomial coefficients is returned instead of evaluated baselines.

`poly.order`: polynomial order of the baselines

`npts.min`: minimal number of data points per spectrum to be used for the fit.
`npts.min` defaults to the smaller of 3 times (`poly.order` + 1) or $\frac{1}{20th}$ of the number of data points per spectrum.
If `npts.min` ≤ `poly.order`, a warning is issued and `npts.min <- poly.order + 1` is used.

`noise`: a vector giving the amount of noise, see below.

`short`, `user`, `date`: are handed to `logentry`

Specifying the spectral range

The polynomials are always fit to `fit.to`, but are evaluated on the the wavelengths of `apply.to`. Therefore it is possible to exclude spectral regions that do not contribute to the baseline from the fitting, while the baseline is used for the whole spectrum. The choice of the spectral range in `fit.to` influences the resulting baselines to a certain extent, as becomes clear from figure 1. Narrowing the spectral range in `fit.to` speeds up the fitting of the polynomials:

```
> system.time(spc.fit.poly.below(chondro[], NULL, poly.order = 1, npts.min = 20,
+     noise = 12))
   user  system elapsed
  2.796   0.008   2.834
> system.time(spc.fit.poly.below(chondro[, , -wl2i(chondro, 700 ~ 1700), wl.index = TRUE],
+     NULL, poly.order = 1, npts.min = 20, noise = 12))
   user  system elapsed
  1.820   0.004   1.829
```

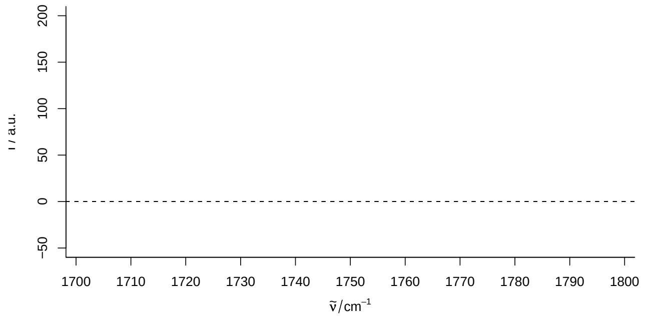

Figure 1: Influence of `fit.to` on the baseline polynomial. The black baseline is fit to the spectral range $1700 - 1800$ cm^{-1}, the blue to $1715 - 1800$ cm^{-1} only (dots & circles: supporting points).

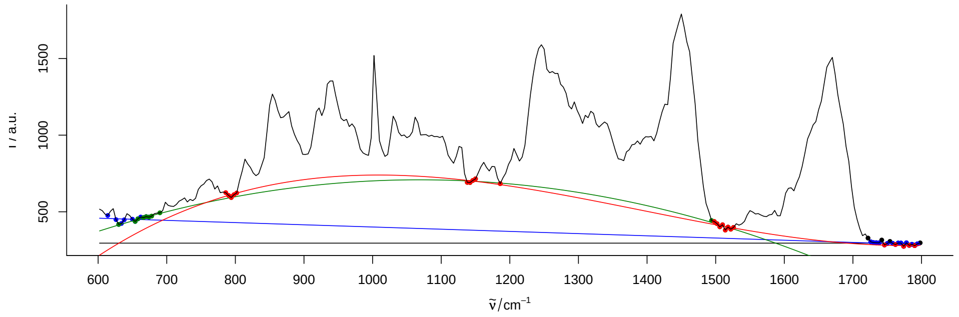

Figure 2: Baseline polynomial fit to the first spectrum of the chondro data set of order 0 (black), 1 (blue), 2 (green), and 3 (red). The dots indicate the points used for the fitting of the polynomial.

Fitting polynomials of different orders

Figure 2 shows the resulting baseline polynomial of `spc.fit.poly.below (chondro [1], poly.order = order)` with `order` = 0 to 3 for the first spectrum of the chondro data set.

The mecanism of fitting the baselines

Appropriate spectral regions with supporting points for the baseline polynomials are calculated iteratively:

1. A polynomial of the requested order is fit to the considered spectral range.
2. Only the parts of the spectrum that lie below this polynomial plus the `noise` are retained as supporting points for the next iteration.

These two steps are repeated until either

- no further points are excluded, or
- the next polynomial would have less than `npts.min` supporting points.

The baselines and respective supporting points for each iteration of `spc.fit.poly.below (chondro [1], poly.order = 1)` are shown in figure 3.

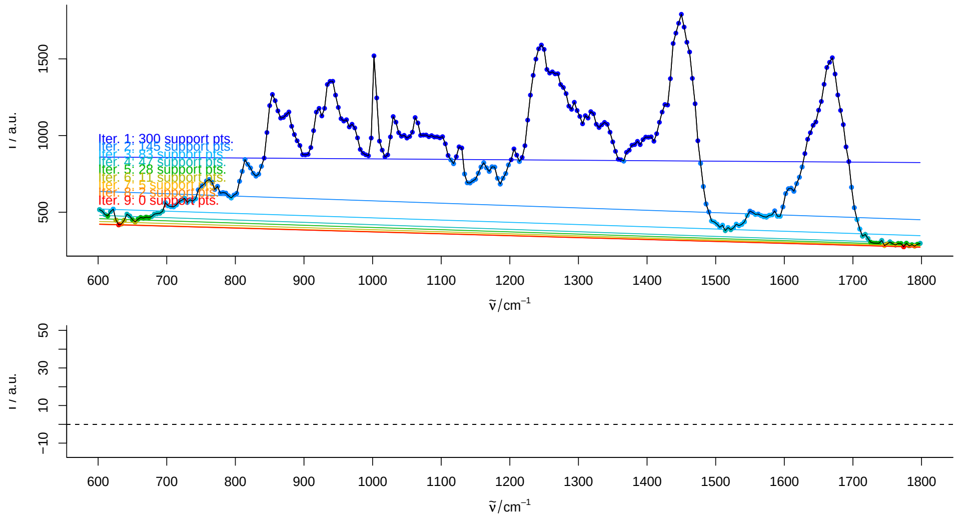

Figure 3: Iterative fitting of the baseline. The dots give the supporting points for the baselines in the same color. The lower part is a magnificaton of the intensity axis.

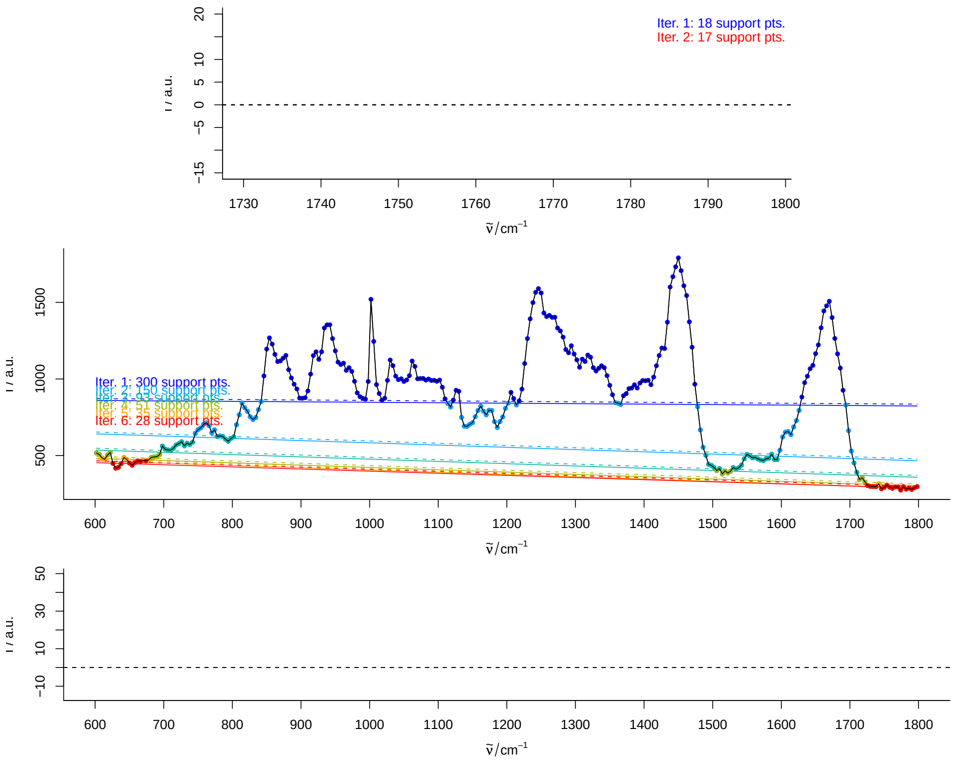

Figure 4: Iterative fitting of the baseline with noise level. Upper part: effects of the noise parameter on the baseline of a spectrum consisting only of noise and baseline: without giving `noise` the resulting baseline (black) is clearly too low. A noise level of 12 results in the red baseline. The middle and lower part show the baseline fitting with noise level on the complete spectrum. Color: iterations, dots/circles: supporting points for the respective baselines. Dashed: baseline plus noise. All points above this line are excuded from the next iteration.

The noise level

Besides defining a minimal number of supporting points, a "noise level" may be given. Consider the spectral range above 1730 cm^{-1} of the shown spectrum. The spectrum consists practically only of noise. The upper part of figure 4 illustrates the problem. As the baseline fitting algorithm cannot distinguish between noise and real bands appearing above the fitted polynomial, the resulting baseline (black) is too low if the `noise` parameter is not given.

Setting the noise level to 12 (roughly 1.5 standard deviations), the fitting converges after 2 iteratons with a much better result. The resulting baselines for `spc.fit.poly.below (chondro [1], poly.order = 1, noise = 12)` of the whole spectrum are shown in the middle and lower part of figure 4

`noise` may be a single value for all spectra, or a vector with the noise level for each of the spectra separately.