2 Spectral data processing

In this session, we perform different (pre-)processing steps that are frequently employed in hyperspectral data analysis. In particular we test narrow band vegetation indices, the Savitzky Golay filter (Savitzky & Golay 1964) as spectral smoothing technique and spectral derivatives as technique to enhance spectral features (e.g. Demetriades-Shah et al. 1990). As test data set we use airborne AVIRIS spectra and leaf nitrogen samples from forest ecosystems in the North-eastern United States provided by the University of Wisconsin through the ecosis.org repository. A detailed data set description is provided in Singh et al (2015).

Importing and screening the data

Download the data in the three files "allplotsaggregatedspectra.txt", "bands.txt" and "N_percDW_all.txt" from Moodle. "allplotsaggregatedspectra.txt" contains the spectra in rows and wavelengths in columns. The original bands 1 and 2 were previously removed from the data set. "bands.txt" contains the wavelength information and an indicator of band quality. "N_percDW_all.txt" provides the leaf nitrogen (in % leaf dry weight) information that was measured for each observation in four replicates. Read the data into your workspace, e.g. in R using

```
dat <- read.table ("allplotsaggregatedspectra.txt", header=T)
wl <- read.table ("bands.txt", header=T,sep=",")
traits <- read.table ("N_percDW_all.txt", header=T)
```

From the leaf nitrogen data, we calculate a mean value per sample as response variable in our analyses.

```
n <- apply (traits, 1, mean)
```

Take a first look at the data by plotting a few of them against the wavelength of the respective bands.

```
plot (wl[,2], dat[1,], type="l",xlab="Wavelength / nm", ylab="Pixel value")
```

The spectra contain several strange features that result from noise in the data. The data providers have hence already identified and flagged bad bands in the wavelength table. These bands are highlighted in the 3rd column (“IsBad”) by a value of “1” (a value “0” indicates a good quality).

```
points (wl[wl[,3]==1, 2], dat[1,wl[,3]==1], pch=19, col="red")
```

Q2.1 (3 pts) In which regions are the bad bands located? What are possible reasons for their quality issues?

Remove all bad bands that are labeled with “1” in the 3rd column from the spectral data set.

```
dat <- dat[,wl[,3]==0]
wl <- wl[wl[,3]==0,2]
```

Further, due to a detector overlap, two wavelengths are replicated in the data set with two bands each. We eliminate the replicates as well.

```
dat <- dat[,-c (91,93)]
wl <- wl[-c (91,93)]
```

Plot again the first spectrum to check the changes in the data.

```
plot (wl, dat[1,], type="l", xlab="Wavelength / nm", ylab="Pixel value")
```

Resampling the spectra to 1 nm resolution

For the upcoming operations, it is useful to increase the spectral resolution of the data to 1 nm. This allows to extract, for example, the reflectance values for specific wavelength for the calculation of narrow band indices.

Q2.2 (1 pt) What is the original spectral resolution of the data?

In R, a simple way to resample spectral data is provided through the approx() function. This function requires a spectrum and the wavelength information as vectors y and x as well as a vector of the desired intervals (xout) for which the reflectance values are interpolated. It returns both the interpolated spectrum ($y) and the new bands ($x). For the sake of applicability, we wrap the approx-function with a simple wrapper.

```
refine <- function (wl, spec, wl.out){ ## wl = x, spec = y, wl.out = xout
  approx (x=wl, y=spec, xout=wl.out)$y ## return only the interpolated spectrum
}
```

We create a vector of the spectral range of the data in 1 nm steps and apply the refine() function on each spectrum (row) in the data set. The result dat1 contains the spectra in interpolated 1 nm resolution.

```
wl1 <- wl[1]:wl[170]
dat1 <- t (apply (dat, 1, refine, wl=wl, wl.out=wl1)) ## t() for back-
                                  ## transposing the result to spectra in rows
```

Plot the first spectrum once again to screen the data.

Narrow band vegetation indices

Several narrow band vegetation indices promise a simple assessment of leaf nitrogen from spectral data. Two of them are the red edge inflection point (REIP, Guyot et al. 1988) and the normalized difference red edge index (NDRE, Barnes et al. 2000). Both indices exploit wavelengths in the NIR and red edge region that are reported to be sensitive to changes in leaf nitrogen. The NDRE is a normalized difference index in the tradition of the NDVI (eq. 2.1). It enhances and normalizes the differences between two wavelengths at the transition from the red edge to the NIR-plateau. The REIP aims to identify the wavelength where the slope of the red edge shows a significant change (eq. 2.2).

$$NDRE = \frac{Ref_{790\,nm} - Ref_{720\,nm}}{Ref_{790\,nm} + Ref_{720\,nm}} \qquad \text{(eq. 2.1)}$$

$$REIP = 700 + 40 * \left(\frac{\frac{Ref_{670\,nm} + Ref_{780\,nm}}{2} - Ref_{700\,nm}}{Ref_{740\,nm} - Ref_{700\,nm}} \right) \qquad \text{(eq. 2.2)}$$

First, calculate the NDRE. My code applies the refine() function again, this time only returning the two wavelengths that are needed for this index. Obviously these wavelengths can likewise be extracted from dat1.

```
dat2 <- t (apply (dat, 1, refine, wl=wl, wl.out=c (720, 790)))
NDRE <- (dat2[,2] - dat2[,1]) / (dat2[,2] + dat2[,1])
```

Second, we calculate the REIP.

```
dat3 <- t (apply (dat, 1, refine, wl=wl, wl.out=c (670, 700, 740, 780)))
REIP <- 700 + 40 * ((((dat3[,1] + dat3[,4]) / 2) - dat3[,2]) / (dat3[,3] -
        dat3[,2]))
```

Q2.3 (1 pt) What are the ranges of the returned index values? Which (statistical) distribution do they have?

Let us see if the indices keep their promise and perform well as indicators of leaf nitrogen.

```
plot (n, NDRE)
cor (n, NDRE)^2
plot (n, REIP)
cor (n, REIP)^2
```

Q2.4 (1 pt) Describe the performance of the two indices as indicators of leaf nitrogen. Which one is apparently more suitable? How do you judge their overall performance?

Q2.5 (3 pts) Leaf nitrogen was measured as percentage of the leaf dry weight. Is this a suitable metric for spectral analyses? Suggest some improvements for future data acquisitions and analyses.

Spectral smoothing

Despite the removal of the bad bands, the spectra still visibly contain a significant amount of noise. We apply the Savitzky-Golay filter to mitigate this noise. The R package ‘signal’ provides a nice implementation of this filter. Install and load this package. If you are using a different language for your analyses, search and find an implementation.

In the signal-package, the filter is implemented in the sgolayfilt() function. This function smoothes a spectral vector with a polynom of a given order (argument p) and for a defined window size (argument n). We apply it spectrum-wise (i.e., row-wise) to the interpolated data set dat 1. As a starting point, we use a 3rd degree polynom and a window size of 131 nm.

```
dat4 <- t (apply (dat1, 1, sgolayfilt, p=3, n=131))
```

Compare the result to the original data and visualize the effects of the smoothing.

```
plot (wl, dat[1,], type="l", xlab="Wavelength / nm", ylab="Pixel value")
lines (wl1, dat4[1,], col=2)
```

Q2.6 (2 pts) How do the degree of the polynom and the window size affect the outcome? Vary the polynom (e.g. 2nd degree, 4th degree, etc.) and the window size and compare the results. Which differences become apparent?

Recalculate the NDRE and the REIP for the smoothed data (using the filter parameters p=3, n=131).

```
REIP2 <- 700 + 40 * ((((dat4[,257] + dat4[,367]) / 2) - dat4[,287]) /
        (dat4[,327] - dat4[,287]))
NDRE2 <- (dat4[,377] - dat4[,307]) / (dat4[,377] + dat4[,307])
```

Q2.7 (1 pt) Repeat the comparison of the index values to the leaf nitrogen values. Does the smoothing change the result? Did the performance of the two indicators improve?

Spectral derivatives

Besides smoothing, the sgolayfilt() function can return the n-th derivative of the spectrum. It hence combines the smoothing ability with the derivatives to enhance the interesting spectral features without increase of the high-frequency noise. The order of the derivative is defined by the argument m (m=0 is a pure smoothing of the spectrum).

```
dat5 <- t (apply (dat1, 1, sgolayfilt, p=3, n=131, m=1)) ## for the 1st order
                                                         ## derivatives
plot (wl1, dat5[1,], type="l", xlab="Wavelength/nm", ylab="1st derivative")
```

The most pronounced (maximum) peak in the derivative corresponds to the steep increase in the red edge, which is the most prominent feature in the present data set. In theory, similar to the REIP the position of this peak indicates the red edge inflection points. We extract the corresponding wavelength and check if they match.

```
mxd1 <- wl1[apply (dat5, 1, which.max)]
```

Q2.8 (2 pts) What is the range of the wavelength position of the 1st order derivative? Compare it to the REIP. Is this position of the 1st derivative maximum in line with the red edge infection point? If not, how do they differ in their assessments?

Smoothing spectral image data

Finally, we test the smoothing operation on image data sets. For this purpose, we go back to the image data sets from session 1. Load the image that you used last time in your work space. Use a spatial subset of 100 pixels x 100 pixels to make the operation run faster (in R and using the terra package first define a smaller extent of 100 px x 100 px with ext() and then use this extent to clip the image with the crop() function). The typical work flow for smoothing image pixels is

1. *extract the pixel spectra into a pixel x band table (pixels in rows, bands in columns)*
2. *if the bands have an irregular wavelength spacing, interpolate to regular intervals*
3. *identify a suitable polygon and window size for the Savitzky-Golay filter that results in the desired level of smoothing*
4. *smooth the spectra in the (interpolated) pixel x band table*
5. *If in step 2 an interpolation was performed, interpolate the smoothed spectra back to the original bands.*
6. *write the smoothed spectra back to the image matrix*

Q2.9 (6 pts) Implement this work flow in your environment. How do you do it? Which polynom and window size did you choose and how did you come to this parameterization? Provide a plot of a smoothed and original pixel spectrum to visualize your results.

Copy your answers to problems Q2.1-Q2.9 to a single document and submit it to the Moodle portal. For each problem, you can get up to the number of points listed in the problem (in total 20 pts). You can earn two extra points for using a different programming language or for deviating from the code that I provided. The grading of your results follows table 2.1. Have fun!

Table 2.1 Grading scheme

Pts	20	19	18	17	16	15	14	13	12	11
Grade	1.0	1.3	1.7	2.0	2.3	2.7	3.0	3.3	3.7	4.0

References

Barnes, E.M., T.R. Clarke, S.E. Richards, P.D. Colaizzi, J. Haberland, M., Kostrzewski, P. Waller, C. Choi, E. Riley, T. Thompson, R.J. Lascano, H. Li, Moran M. S. (2000). Coincident detection of crop water stress, nitrogen status and canopy density using ground-based multispectral data [CD ROM]. In P.C. Robert et al. (ed.) Proc. Int. Conf. Prec. Agric., 5th, Bloomington, MN. 16–19 July 2000. ASA, CSSA, and SSSA, Madison, WI.

Demetriades-Shah, T.H., Steven, M.D., Clark, J.A. (1990). High resolution derivative spectra in remote sensing. Remote Sensing of Environment 33, 55-64.

Guyot, G., Baret, F., Major, D. J. (1988). High spectral resolution: Determination of spectral shifts between the red and infrared. International Archives of Photogrammetry and Remote Sensing, 11, pp. 750–760.

Savitzky, A., Golay, M.J.E. (1964). Smoothing and differentiation of data by simplified least squares procedures. Analytical Chemistry 36, 1627-1639.

Singh, A., Serbin, S.P., McNeill, B.E., Kingdon, C.C., Townsend, P.A. (2015). Imaging spectroscopy algorithms for mapping canopy foliar chemical and morphological traits and their uncertainties. Ecological Applications 25, 2180-2197.