Introduction to spectral processing with 'spectrolab'

Anna K. Schweiger

Install spectrolab

```
install.packages("spectrolab")
```

```
library("spectrolab")
```

The spectra class

`spectrolab` defines a new S3 class called `spectra` that holds all of the different components of a spectral data.

Without diving too much into its implementation, a `spectra` object holds the important information needed for most spectral data sets: reflectance, wavelengths, filen ames, metadata etc. The class has a bunch of requirements in terms of both format and values.

Read and inspect data

Our spectral data were measured with an instrument called *ASD*. '

```
spec <- read_spectra("./example_data/", format="asd")
```

`spectrolab` can also read other file formats, but let's not worry about that for now. You can always look at `spectrolab's` help for more information.

```
help(read_spectra)
```

You can see what a `spectra` object contains by typing

```
spec
```

```
## spectra object
## number of samples: 9
## bands: 350 to 2500 (2151 bands)
## metadata: none
##
##                350                351                352
## 103_SALHU00000 0.0553291944884881 0.044910506095899  0.0523716595312952
## 103_SALHU00001 0.0765274048215119 0.0717495824057209 0.0696882043408338
```

```
## 103_SALHU00002 0.0685952189080724 0.0611274666766303 0.0582451955497961
## 103_SPITO00000 0.0771597809774897 0.0708588590589189 0.0699624301626821
## 103_SPITO00001 0.059278354402922  0.0561048473944045 0.0479089283439103
##                353                354                355
## 103_SALHU00000 0.0492809575754814 0.0450551981658801 0.0472716432998146
## 103_SALHU00001 0.065932651726352  0.0641965595587686 0.0658921140029544
## 103_SALHU00002 0.0635829576472747 0.062906334767177  0.0529386103808933
## 103_SPITO00000 0.0683834918526438 0.0603142651099755 0.0496831080863538
## 103_SPITO00001 0.0404531509557369 0.0419312955805392 0.051260307662551
##                356                ...
## 103_SALHU00000 0.048303546854188
## 103_SALHU00001 0.0658540579024284
## 103_SALHU00002 0.0553967379235214
## 103_SPITO00000 0.0592112258665227
## 103_SPITO00001 0.0502698726818367
```

```
str(spec)
```

```
## List of 4
##  $ value: num [1:9, 1:2151] 0.0553 0.0765 0.0686 0.0772 0.0593 ...
##  $ bands: num [1:2151] 350 351 352 353 354 355 356 357 358 359 ...
##  $ names: chr [1:9] "103_SALHU00000" "103_SALHU00001" "103_SALHU00002" "103_SPITO00000" ...
##  $ meta :'data.frame':   9 obs. of  0 variables
```

To access the metadata just type

```
meta(spec)
```

```
## data frame with 0 columns and 9 rows
```

Okay, our metadata st

#... and plot it plot(spec) plot_quantile(spec,add=T) # plot_interactive(spec)

‘‘‘