

# Matching sensor overlap

Jose Eduardo Meireles

## The problem

Instruments may measure different spectral ranges using different sensors, resulting in abrupt “jumps” in the reflectance or radiance data. In such cases, the regions between sensors need to be matched, i.e. spliced together. Unmatched spectra collected with a 3-sensor instrument, such as the SVC HR1024, may look like this:

```
# Path to raw (unmatched) spectra
path_raw = system.file("extdata/svc_raw_and_overlap_matched_serbin/SVC_Files/",
                        package = "spectrolab")

# Read spectra as reflectance and radiance
reflect_raw = read_spectra(path = path_raw, type = "target_reflectance")

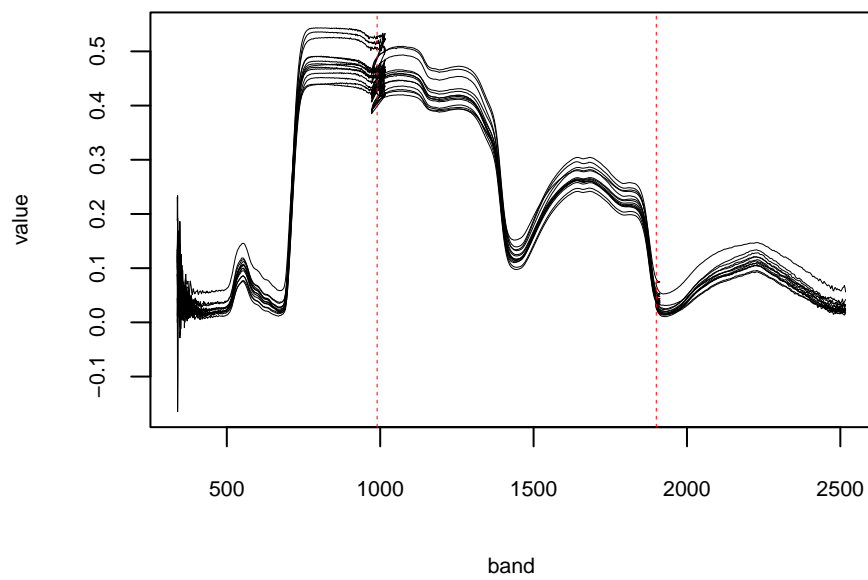
radiance_raw = read_spectra(path = path_raw, type = "target_radiance")

# Sensor overlaps marked with vertical dashed lines
lwd = 0.5
cex = 0.7
par(mfrow = c(2, 1))

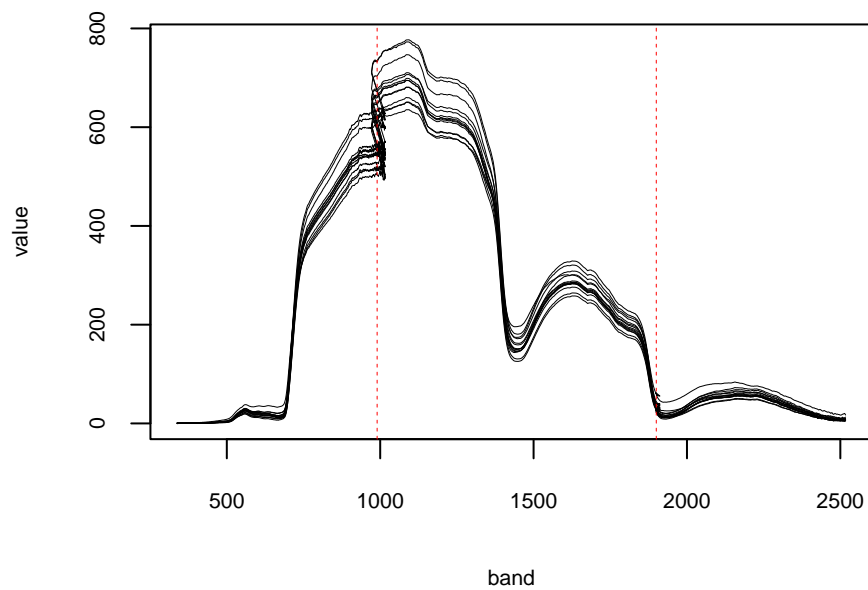
plot(reflect_raw, main = "Reflectance",
     lwd = lwd, cex.main = cex, cex.lab = cex, cex.axis = cex)
abline(v = c(990, 1900), col = "red", lty = 2, lwd = lwd)

plot(radiance_raw, main = "Radiance",
     lwd = lwd, cex.main = cex, cex.lab = cex, cex.axis = cex)
abline(v = c(990, 1900), col = "red", lty = 2, lwd = lwd)
```

**Reflectance**



**Radiance**



## The solution

Use the function `match_sensors` to splice the sensor overlap regions. You must pass the boundary between sensors using the `splice_at` argument. It is **critical** that you get those bands right (or very close) and every instrument (even from the same vendor) is different.

You can use `plot_interactive` zoom into a particular spectral region and decide what the `splice_at` values should be. You can also use the function `guess_splice_at` to estimates these bands for you but keep in mind that these are **guesses**.

```
# Spectrolab's guess of what the splice bands are.
# However, you should also visually inspect the spectra to determine what the
# boundaries between sensors are.
splice_bands_guess = guess_splice_at(reflect_raw)
splice_bands_guess

## [1] 986.7333 1901.1000

# Finally, if you know what those sensor bounds should be (say, they're given by
# the manufacturer), just use those numbers instead of spectrolab's guess.
splice_bands = c(990, 1900)

# Match the reflectance and radiance data
reflect_matched = match_sensors(x = reflect_raw, splice_at = splice_bands,
                                interpolate_wvl = c(5, 1))

radiance_matched = match_sensors(x = radiance_raw, splice_at = splice_bands,
                                  interpolate_wvl = c(5, 1))

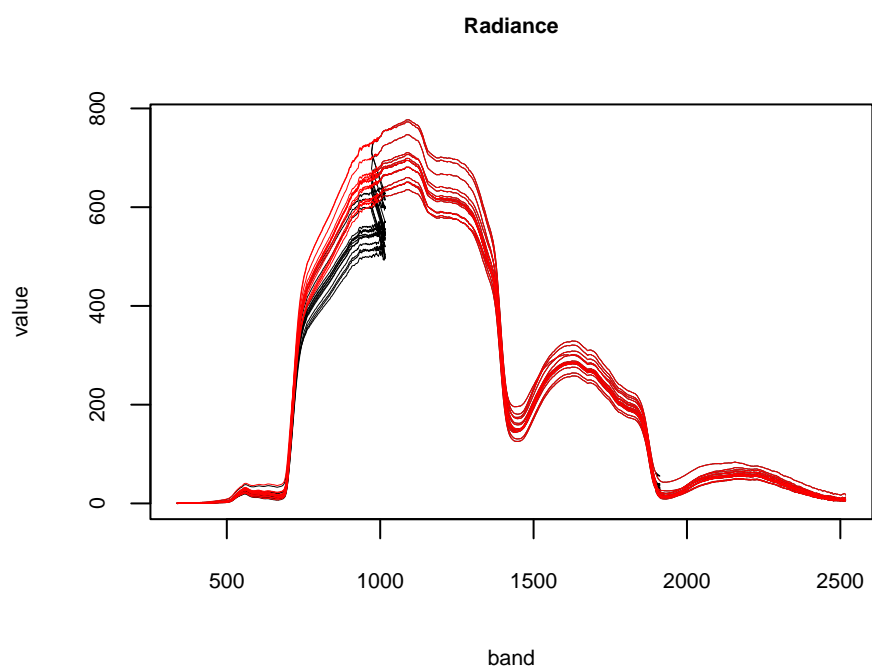
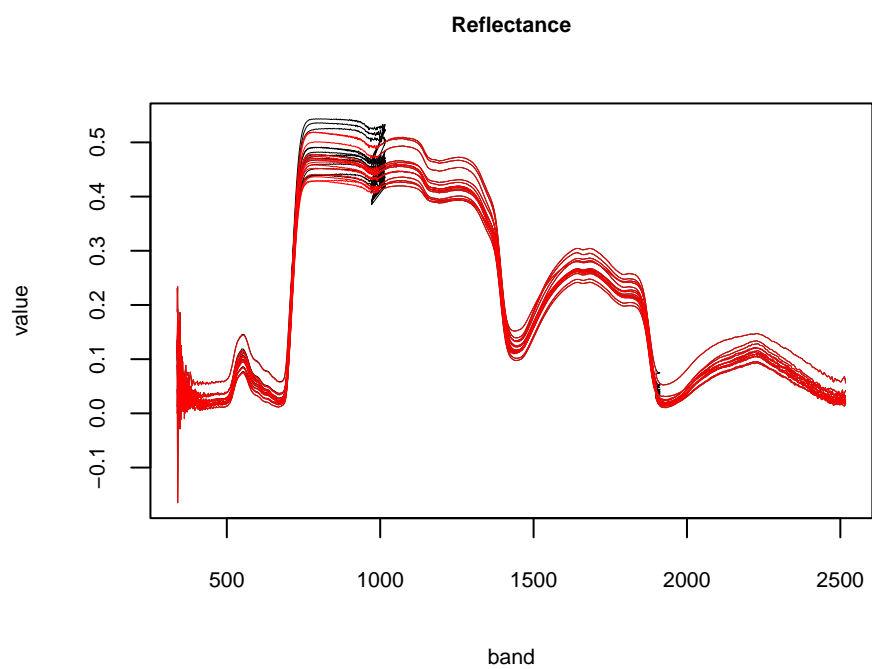
lwd = 0.5
cex = 0.7
par(mfrow = c(2, 1))

plot(reflect_raw, main = "Reflectance",
     lwd = lwd, cex.main = cex, cex.lab = cex, cex.axis = cex)

plot(reflect_matched, col = "red", add = TRUE,
     lwd = lwd, cex.main = cex, cex.lab = cex, cex.axis = cex)

plot(radiance_raw, main = "Radiance",
     lwd = lwd, cex.main = cex, cex.lab = cex, cex.axis = cex)

plot(radiance_matched, col = "red", add = TRUE,
     lwd = lwd, cex.main = cex, cex.lab = cex, cex.axis = cex)
```



And we can check the results from spectrolab's `match_sensors` against SVC's proprietary matching algorithm.

```

path_moc = system.file("extdata/svc_raw_and_overlap_matched_serbin/SVC_Files_moc/",
                        package = "spectrolab")

reflect_moc = read_spectra(path = path_moc, type = "target_reflectance")

radiance_moc = read_spectra(path = path_moc, type = "target_radiance")

lwd = 0.5
cex = 0.7
par(mfrow = c(2, 1))

plot(reflect_moc, main = "Reflectance", col = "black",
     lwd = lwd, cex.main = cex, cex.lab = cex, cex.axis = cex)

plot(reflect_matched, col = "red", add = TRUE,
     lwd = lwd, cex.main = cex, cex.lab = cex, cex.axis = cex)

plot(radiance_moc, main = "Radiance", col = "black",
     lwd = lwd, cex.main = cex, cex.lab = cex, cex.axis = cex)

plot(radiance_matched, col = "red", add = TRUE,
     lwd = lwd, cex.main = cex, cex.lab = cex, cex.axis = cex)

```

