HSItools documentation

Maurycy Żarczyński

Invalid Date

Table of contents

Preface

This is a companion book to the HSItools R package, aiming at processing and visualizing hyperspectral scanning data.

Maurycy Żarczyński, David C. Edge, Nick P. McKay, and Paul D. Zander developed the package with the community's help.

The current requirements to run HSItools are as follows:

R: 4.1 – this is necessary because we depend on the native R pipe and lambda functions introduced with R 4.1

1 Initial state

HSItools offers an easy way to preprocess Specim data. However, if the data follows the same rules, it can be generalized to the broader workflow.

1.1 Data structure

2 Shiny app

Our shiny app allows quick choice of data, settings, regions of interest (ROI), and depth calibration. Here, we walk through the entire app, screen by screen.

2.1 Screen 1: Initial settings

On this screen, you have to make an initial choice. First, you need to decide whether you need to normalize data or not. If you have a reflectance file from other software, like Lumo®, you probably do not need to normalize the file from the beginning. This is one of the most time-consuming processes. Suppose you decide that you need to normalize your data. In that case, you can select other integration times for your white and dark references if you scanned your target with different settings for target and references. This can happen if you were worried about white reference overexposure, whereas your target was very dark. Finally, you can select some proposed HSI indices from the defaults.

2.2 Screen 2: Data choice

2.3 Screen 3: Cropping

2.4 Screen 4: ROI selection

2.5 Screen 5: Calibration

3 Preprocessing

3.1 Normalization

Before any spectral indices and properties are calculated, normalizing the data and expressing it as a reflectance is necessary. Here, reflectance is a fraction of the signal between the dark and white references acquired during or before the scan.

Normalization is achieved by following the equation:

Which can be modified for different acquisition setups for references:

3.1.1 Normalization with Shiny output

If Shiny GUI was used for data selection, cropping, and calibration, then it is easy to pass the output of Shiny to the normalization routine. The normalized file or files will be written into your data's products directory.

```
# Create normalized reflectance file
reflectance <- hsi_tools_core |>                                      ①
  HSItools::prepare_core()                                           ②
```

① The Shiny GUI output.
② Normalization function.

It is possible to iterate over multiple directories at once using the `{purrr}` package.

3.1.2 Normalization of the directory (no Shiny output)

If no Shiny output is available and input is not produced by hand, the normalization routine can be run without it. In such a case, the entire capture data will be normalized. However, without Shiny output, it is harder to calibrate distances properly.

```
# Create normalized reflectance file
reflectance <- hsi_tools_core |>
  HSItools::prepare_core()
```

It is possible to iterate over multiple directories at once using the `{purrr}` package.

4 Summary

In summary, this book has just begun.