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 Indices

HSItools offers a way to calculate a few of the most common indices classes used in paleoenvironmental investigations.

4.1 Mean reflectance (Rmean)

The most straightforward index is the mean of the reflectance values of all selected wavelengths within a given pixel. It reflects overall changes in the darkness or brightness of the captured specimen.

```
rmean <- reflectance |>                                                        ①
  HSItools::calculate_rmean()                                                  ②
```

① Reflectance data is either in memory or from a disk.
② Calculation function.

4.2 Relative Absorption Band Depth (RABD)

A common index which

The RABD calculation has variations, but the results are generally not drastically different. You can use predefined values or provide them manually.

The output's name informs you about the calculated proxy and additional modifications to the reflectance file; here, we see that it was calculated with Savitzky-Golay smoothed reflectance. Let's calculate one of the most common indices to estimate the total chloropigments-*a*: $\text{RABD}_{660:670}$.

4.2.1 Variant 1 – “max”

In this variant, a minimum reflectance is found in the trough for each pixel and flexibly used for calculations.

```
rabd_max <- reflectance |>                                                ①
  HSItools::calculate_rabd(                                               ②
    edges = proxies$rabd_b660b670$edges,                                  ③
    trough = c(660:670),                                                  ④
    rabd_name = proxies$rabd_b660b670$proxy_name,                         ⑤
    rabd_type = "max")                                                    ⑥
```

① Reflectance data is either in memory or from a disk.
② Calculation function.
③ Edges of the trough, the broader scope.
④ Trough of interest, a narrower scope to find the reflectance value.
⑤ Name of the index to be stored in the raster data.
⑥ Type of the RABD calculation.

4.2.2 Variant 2 – “strict”

This classic variant supplies a specific wavelength to calculate RABD for every pixel.

```
rabd_strict <- reflectance |>
  HSItools::calculate_rabd(
    edges = proxies$rabd_b660b670$edges,
    trough = proxies$rabd_b660b670$trough,
    rabd_name = proxies$rabd_b660b670$proxy_name,
    rabd_type = "strict")
```

4.2.3 Variant 3 – “midpoint”

This is variant 2 (strict), with the added shortcut of always finding the middle point between the through edges - a convenience shortcut for some.

```
rabd_midpoint <- reflectance |>                                           ①
  HSItools::calculate_rabd(                                               ②
    edges = proxies$rabd_b660b670$edges,                                  ③
    trough = proxies$rabd_b660b670$trough,                                ④
    rabd_name = proxies$rabd_b660b670$proxy_name,                         ⑤
    rabd_type = "mid")                                                    ⑥
```

4.3 Relative Absorption Band Area

4.4 Spectral ratios

Another popular and straightforward index is band ratios, where reflectance at wavelength X is divided by reflectance at wavelength Y.

```
ratio_570630 <- reflectance |>                              ①
  HSItools::calculate_band_ratio(                          ②
    edges = proxies$ratio_b570b630$edges,                  ③
    ratio_name = proxies$ratio_b570b630$proxy_name)        ④
```

① Reflectance data is either in memory or from a disk.
② Calculation function.
③ Wavelength X and Y, numerator and denominator.
④ Name of the index to be stored in the raster data.

4.5 Derivatives

4.6 Differences

```
difference_650675 <- reflectance |>                         ①
  HSItools::calculate_band_difference(                     ②
    difference_name = proxies$diff_b650b675$proxy_name,    ③
    edges = proxies$diff_b650b675$edges)                   ④
```

① Reflectance data is either in memory or from a disk.
② Calculation function.
③ Wavelength X and Y.
④ Name of the index to be stored in the raster data.

4.7 Other

4.7.1 Red Edge Minimum Point (REMP)

This index was introduced in the paper “Ghanbari, H., Zilkey, D.R., Gregory-Eaves, I., Antoniades, D., 2023. A new index for the rapid generation of chlorophyll time series from hyperspectral imaging of sediment cores. Limnology and Oceanography: Methods 21, 703–717.

```r
remp <- reflectance |>                                                    ①
  HSItools::calculate_remp()                                              ②
```

5 Summary

In summary, this book has just begun.

References