Fractional Coverage Algorithm: AutoMCU

Authors: Elahe Jamalinia, Nick Vaughn
Drafted: Feb 2, 2023
Last Updated: Feb 28, 2023

Table of Contents

1. Introduction

Carbon Mapper is an airborne and satellite-based hyperspectral mission to provide data for greenhouse gas emissions, and land and ocean applications. One of the areas that Carbon Mapper Land and Ocean products will contribute to is the quantification of non-photosynthetic vegetation (NPV) in agricultural lands (G. Asner et al., 2022). Monitoring of both photo-synthetic (green) vegetation (PV), and NPV is important to understand a range of ecosystem characteristics including vegetation presence, fractional cover, physiological and biogeochemical functioning, drought severity, disturbance events and recovery from disturbance. NPV refers to the vegetation that cannot perform photosynthesis, which mainly includes woody stems and below-ground dead biomass and dormant vegetation. Ecological and biogeochemical research at the landscape and regional scale requires detailed information on the spatial and temporal variability of live vegetation, dry carbon, and bare soils. On the global scale, NPV is a large carbon pool in natural ecosystems. Thus, quantifying NPV is important for understanding carbon sequestration, which is a strategy to lower atmospheric CO_2 concentration. In addition, NPV as wildfire fuel (Kim et al., 2009) also has a potential impact on the climate because the burning of biomass substantially

contributes to global greenhouse gas, aerosol, and black carbon emissions. It also affects the mass and energy balance within the root zone through altering evapotranspiration and water infiltration into the root zone.

Using imaging spectroscopy, NPV fraction can be estimated using spectral unmixing approaches, as NPV significantly contributes to canopy reflectance (G. P. Asner & Heidebrecht, 2003; Huemmrich & Goward, 1997). NPV has much lower chlorophyll and water content than photosynthetic (green) vegetation, while the high cellulose and lignin content in NPV is absent in bare soil (G. P. Asner & Lobell, 2000; Asner Gregory & Lobell David, 2000; Gregory P. Asner and Kathleen B. Heidebrecht, 2000; Numata et al., 2008; Serbin et al., 2013). These differences cause variations in their spectral reflectance, which may be used to estimate the relative fractional coverage of NPV, PV and bare soil.

2. Scope

Here, we describe an approach for using a methodology called Automated Monte Carlo Unmixing (AutoMCU) to quantify the amount of fractional cover of NPV with visible-to-shortwave infrared (VSWIR) imaging spectrometer data. In this document, we describe the AutoMCU algorithm, its performance, and how it can be utilized to differentiate NPV over plant and soil dominated landscapes. The algorithm is demonstrated on 5nm spectral resolution VSWIR data from the Global Airborne Observatory (GAO; CITE) at both extremely high resolution (60 cm) and downsampled in a way to mimic output from the Tanager sensors that are part of the Carbon Mapper satellite program (30 m). While the algorithm can be expanded for more general usage of differentiating other classes, the focus here is on the specific application of AutoMCU to estimate the fractional coverage of the three classes: Photosynthetic Vegetation (PV), Bare Soil (BS), and Non-Photosynthetic Vegetation (NPV), which can be used as a quantifier of surface carbon stored in NPV.

3. AutoMCU

3.1 Algorithm Description

AutoMCU uses a Monte-Carlo resampling approach to quantify the fractional coverage of specified target constituents in reflectance spectra measured with an imaging spectrometer. While the algorithm can be used for other targets, here we aim to identify the fractional cover of PV, NPV, and BS in the observed spectra. These observed spectra, $\rho(\lambda)$, are assumed to be a (linear) mixtures of these classes of the form of Equation 1.

$$\rho(\lambda) = f_{PV}\rho(\lambda)_{PV} + f_{NPV}\rho(\lambda)_{NPV} + f_{BS}\rho(\lambda)_{BS} + \varepsilon(\lambda) \quad (1)$$

Here f_x are the fraction coefficients that should sum to nearly 1.0, $\rho(\lambda)_x$ are the input class example constituent spectra along wavelength λ, and $\varepsilon(\lambda)$ is an error term. Rather than unmixing

a single set of example spectra for each of the three classes, AutoMCU uses a random permutation approach to run SMA many times for each observed spectrum. The algorithm is given a library of example spectra for each class, and it runs SMA many times based on randomly permuted combinations of these endmember classes. For each of *m* user-specified iterations, the algorithm selects a new random combination of one endmember spectra from each of the endmember library bundles to unmix the image spectrum.

Rather than use the entire VSWIR spectral window (~400-2500nm) for unmixing, AutoMCU uses only two regions known to strongly differ between the PV, NPV, and BS classes (Figure 1): 1) the SWIR2 (2000-2300 nm) region is dominated by foliar water absorption in PV spectra, whereas drier NPV vegetation allows the absorption features of organic compounds like cellulose and lignin to stand out at 2100 and 2300 nm. In contrast, BS spectra typically have a hydroxyl absorption feature at 2200 nm. 2) the visible to NIR range (600-800 nm) contains red edge feature of green vegetation. In general, reflectance decreases with increasing organic matter and/or water content in vegetation or soil. Due to the unique spectral shape of green vegetation in the red edge, this region provides a good separation between PV from both NPV and BS (G. P. Asner & Lobell, 2000).

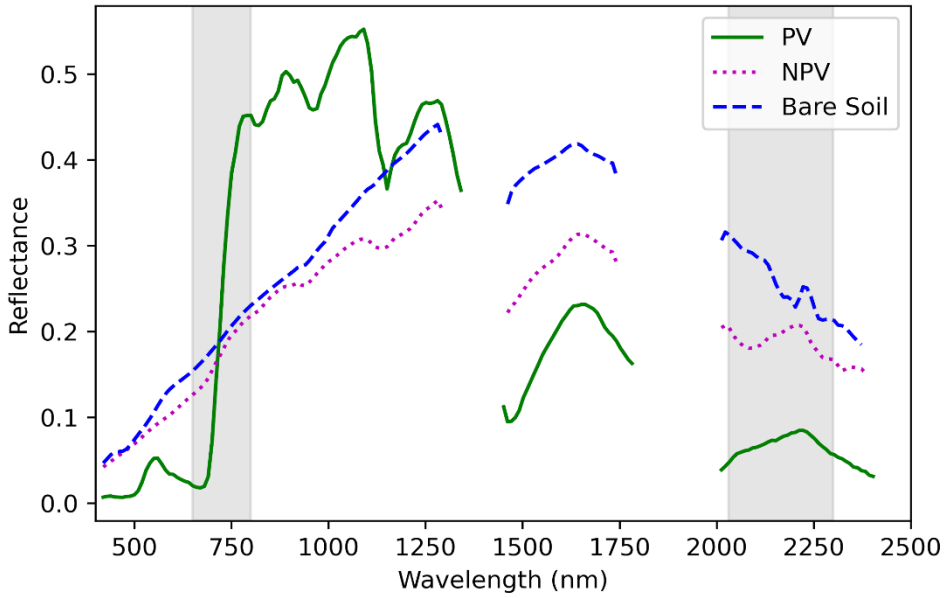


Figure 1. Example spectra for the three classes separated by the AutoMCU algorithm. Band windows used in unmixing are shown in gray.

To reduce the effects of brightness changes across an input image, some form of regularization is needed to ensure that the example spectra for each class can properly fit the observed spectra. To do a traditional brightness normalization, where spectra are divided by some measure of total brightness such as the vector norm, would destroy the linearity assumption of SMA. Instead, in AutoMCU all spectra are "tied" to maintain this linearity. To tie the spectra, the reflectance value of the first entry in each fitting window (red edge, SWIR2) is subtracted from all

reflectance values within that window (Figure 2). This zeroes out the first value of each window but standardizes the remaining spectra to stabilize PV retrievals across variations in biomass, canopy architecture, and biochemistry.

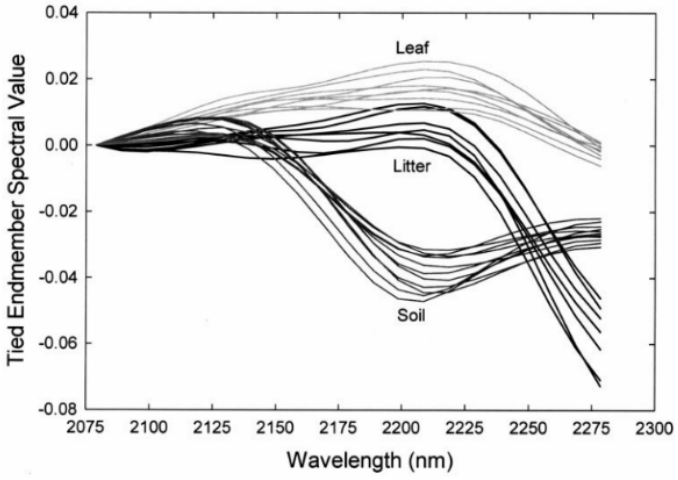


Figure 2. Example of tied spectra within the SWIR2 band window used by the AutoMCU algorithm. Taken from (CITE PAPER here).

After all iterations are complete, the estimated fractions for each input image pixel are determined by computing the trimmed mean endmember fractions for this pixel across all iterations. Two measures of uncertainty are also saved: 1) the standard deviation (std dev) of the fractions across iterations, and 2) the RMSE derived by comparing the model-estimated mixture spectra using the averaged coefficients against the observed spectrum for each input image pixel.

3.2 AutoMCU Performance

AutoMCU has been extensively tested and well vetted over the 20+ years it has been available. The first examination was presented along with the original work introducing the algorithm (Gregory P. Asner and Kathleen B. Heidebrecht, 2000). Here, the authors tested the algorithm against fractional cover estimated with high-resolution airborne photo imagery against fractions estimated using AVIRIS used in AutoMCU algorithm. Results from using the SWIR2 fitting region (no second window with red edge) only show a general agreement with the observed cover within one standard deviation (Table 1).

Table 1 Summary of estimated photosynthetic vegetation (PV), non-photosynthetic vegetation (NPV), and bare soil cover fractions for two different landscapes using AutoMCU on just the SWIR2 fitting window on Aviris (first gen.) airborne spectrometer data. Fractional cover results from field measurements are also given (adapted from Gregory P. Asner and Kathleen B. Heidebrecht, 2000).

Permutation	Class	Shrubland	Std. Dev.	Grassland	Std. Dev.
Observed	PV	0.181	0.010	0.069	0.033

	NPV	0.010	0.010	0.367	0.133
	Soil	0.820	0.020	0.586	0.173
AutoMCU AVIRIS	PV	0.148	0.062	0.129	0.057
	NPV	0.013	0.034	0.326	0.038
	Soil	0.839	0.050	0.548	0.060

In a follow-up study (G. P. Asner & Heidebrecht, 2003) evaluated the AutoMCU algorithm with AVIRIS data at 4m and 30m spatial resolution against field measurements in Argentina drylands. The algorithm produced highly accurate estimates of PV, NPV, and bare soil in both resolutions—R^2 values within 0.83-0.86 range for 4m spatial resolution and within 0.67-0.68 for 30m resolution (Figure 3).

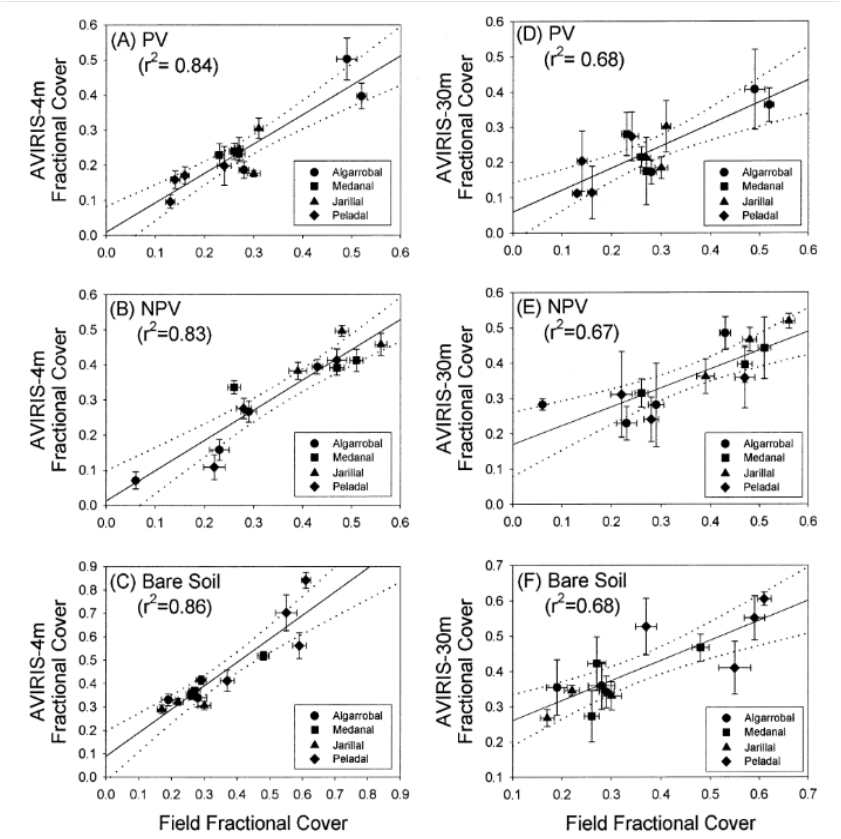


Figure 3. Results from a test of AutoMCU when run on AVIRIS data over Argentinian drylands. Taken from Asner & Heidebrecht, 2003.

A test of AutoMCU against traditional SMA and Multiple-Endmember Spectral Endmember Analysis (MESMA) for identifying fractional cover of PV and NPV using multispectral data (GF-1 Wide View) showed that AutoMCU outperformed the other two methods in for both classes (Li et al., 2016).

4. Benchmark Data

This AutoMCU algorithm was tested using data collected by the GAO VSWIR spectrometer over agricultural farmland in Iowa in July of 2022. This orthorectified image is 8862 rows by 684 columns and has a very high spatial resolution of 0.6 m. The GAO VSWIR data has been atmospherically corrected with ACORN (Atmospheric CORrection Now, Analytical Imaging and Geophysics, LLC), and contains 428 bands with wavelengths ranging from 347-2485 nm with 5 nm spectral resolution (Figure 4a).

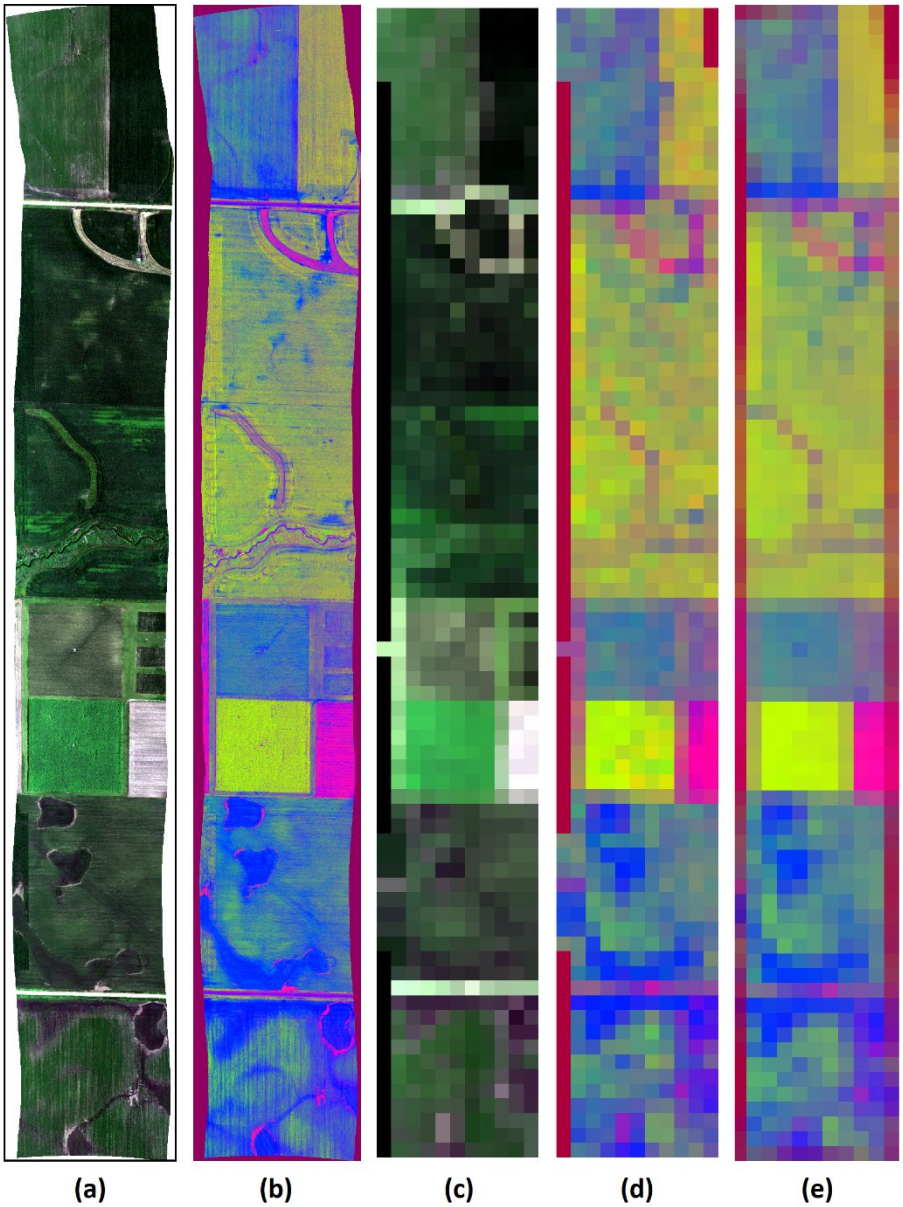


Figure 4. An example high-resolution reflectance image from the GAO VSWIR spectrometer, at full resolution and at Tanager-like resolution, and the AutoMCU results when applied to these data. The maps are a) the full-res RGB representation of the input image, b) false-color representation of the AutoMCU output image (Red= NPV, Green=PV, Blue=BS), c) rescaled reflectance map, d) AutoMCU results when applied to rescaled input map, and e) the full-res AutoMCU results rescaled to 30 m spatial resolution.

In addition, we applied the algorithm to rescaled data designed to mimic that which would be available from the Tanager sensors. First, the radiance data were spectrally convolved to match the slightly different 5 nm band centers ranging from 380 to 2510 nm with 8 nm full width half-max. Then, the data were convolved in raw image space to match the planned 30 m GSD of Tanager. Because the convolution essentially removes random noise from the observed spectra, a random permutation of spectral noise was added to each pixel using a radiometric model provided by Planet Labs PBC. Finally, the data atmospherically corrected and re-orthorectified (Figure 4c).

The AutoMCU algorithm was applied to both the full resolution VSWIR reflectance data and to the Tanager-scale reflectance data (Figure 4b,d). As a test of the algorithm's ability to decipher constituent fractions at the coarser scale, we also spatially resampled the full-resolution AutoMCU output map to 30 m resolution to match the grid of the coarser reflectance data (Figure 4e). The latter was not coarsened in the raw image space like the former, so some spatial misalignment is expected from the difference in orthorectification, but if spatial resolution has little effect on the algorithm, then the two coarse AutoMCU output maps should agree. We found that the maps agree (Figure 5), and thus see no issues when applying this algorithm to actual Tanager data.

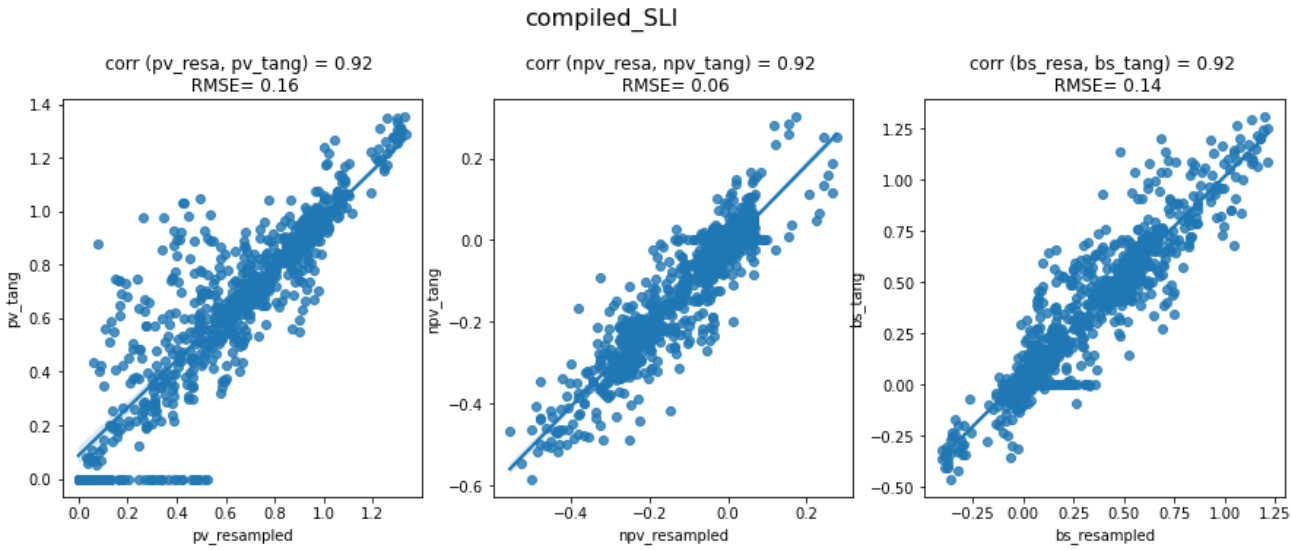


Figure 5. Scatter plots of results from a test of AutoMCU run on coarsened (30 m) reflectance data against the full resolution AutoMCU results coarsened to 30 m for a) PV, b) NPV, and c) BS. There is strong agreement between the outputs despite the high potential for spatial mismatch from the two different methods of coarsening.

5. Codebase

The python package to run AutoMCU is located in the GitHub repository at https://github.com/CMLandOcean/AutoMCU. The Python package is a translation of the original AutoMCU written and developed in C (G. P. Asner & Heidebrecht, 2003; G. P. Asner & Lobell, 2000). Once the package is installed, then running AutoMCU on a single image is handled in the command line using the script amcu_cli.py. Alternatively, the user can run the code from within the amcu subfolder.

5.1 Key Dependencies

The Python version used in testing was 3.9.6, but any version of Python above 3.6 should work as long as it supports modern versions of rasterio. Table 2 lists information about the used packages.

Table 2 Python packages needed for running the AutoMCU.

Package	Version	Repository
numpy	1.22.3	https://github.com/numpy/
pandas	1.4.2	https://github.com/pandas-dev/pandas
rasterio	1.1.0	https://github.com/rasterio
spectral	0.22.4	https://github.com/spectralpython
setuptools	67.2.0	https://github.com/pypa/setuptools
pydantic	1.9.0	https://github.com/pydantic/pydantic
pytest	7.1.2	https://docs.pytest.org/en/latest/
tqdm	4.64.0	https://github.com/tqdm/

5.2 Inputs

The script is highly configurable, but the essential inputs are:

- A separate endmember library for each of the *n* classes (Here, *n*=3: PV, NPV, BS).
  These should be in the form of a CSV with a header row and samples listed in columns, bands as rows, the first column is the wavelength values. The first column must contain wavelength in nanometers. An optional second column containing full width half max is needed if the image

wl	fwhm	Samp001	Samp002	Samp003	Samp004	Samp005
409.89	11.51	0.02763	0.027036	0.026701	0.026998	0.023579
419.9	11.49	0.030562	0.029913	0.029558	0.029929	0.026189
429.92	11.47	0.034817	0.034085	0.033641	0.034121	0.029967
439.93	11.45	0.038906	0.038097	0.037548	0.038142	0.03362
449.95	11.43	0.040886	0.040049	0.039482	0.040164	0.035476
459.96	11.41	0.040692	0.03988	0.039382	0.040135	0.03547
469.98	11.4	0.040505	0.039709	0.039276	0.040109	0.035463
479.99	11.38	0.042686	0.041841	0.041365	0.042319	0.037505

and endmember libraries differ in band centers. All following columns represent the sample spectra for the class. An example of the spectral library is shown below:

- An input image containing VSWIR spectrometer data with a spectral resolution fine enough to differentiate the classes of interest (5 to 10 nm band spacing suggested) in ENVI format (with band wavelength and fwhm information specified if spectral resampling is needed to spectrally match to the library spectra.
- A specification of the wavelength regions that AutoMCU uses for spectral unmixing.
- The number of iterations, *m*, for the algorithm to run for each pixel

5.3 Output

The script will write out an image containing 2**n* + 1 bands, where *n* is the number of endmember classes + 1 for *shade* if specified. The first *n* bands will be the mean fractional cover coefficients of the classes in the specified order. The next *n* bands will be the standard deviations of the fractional cover coefficients, and the last band is the computed RMSE between the modelled spectrum and the observed spectrum. Output can be formatted in any rasterio-supported format, but default is GeoTiff. Band descriptions will be written to the metadata for the output image (i.e., tags in Geotiff or the hdr file for ENVI format).

5.4 Usage

The AutoMCU algorithm runs using command line arguments (an example shows as below), which are divided into required and optional arguments as described below (full command usage is available with the -h flag).

```
python amcu_cli.py REFL PVCSV NPVCSV BSCSV
 --wl_range "650,800" --wl_range "2030,2300" --names "PV,NPV,Bare"
 --iterations 50 --scale 10000 --sum_to_one --emfwhm --nointerp
```

Where:

REFL	The input reflectance image
PVCSV, NPVCSV, BSCSV	CSV-formatted spectral library files
--wl_range "From,To"	Specification of a fitting window, at least 1 needed
--names "PV,NPV,Bare"	Names that will be used in band descriptions of output
--iterations INT	How many MC iterations to run for each pixel
--scale FLT	Multiplier to convert image values to reflectance scaled 0-1
--sum_to_one	Use an extra column of 1s to coerce mixtures to sum to 1.0
--emfwhm	Inform script that second column of input CSVs is fwhm
--nointerp	If input and library spectra are aligned spectrally same number of bands and band centers, this will skip resampling.

5.5 AutoMCU sensitivity to iterations

To test how the number of iterations impacts the result over the example input image, AutoMCU was tested for different numbers of iterations: 30, 50, 70, 100. The input image is the simulated Tanager data and all settings are the same except for iteration number. The influence of number of iterations on the range of the fractions for each end member is negligible (Figure 6). In another try, the spatial difference between the end member fractions for the lowest number of iterations (i.e. 30) and highest number of iterations (i.e. 100) is shown in Figure 7.

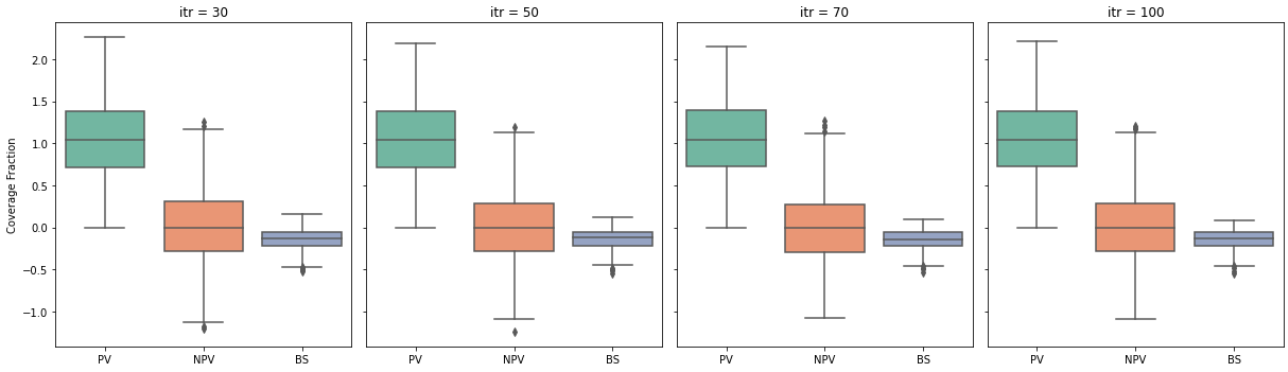


Figure 6. Effect of different number of iterations on AutoMCU results for 3 end member fractions, for 30, 50, 70 and 100 iterations

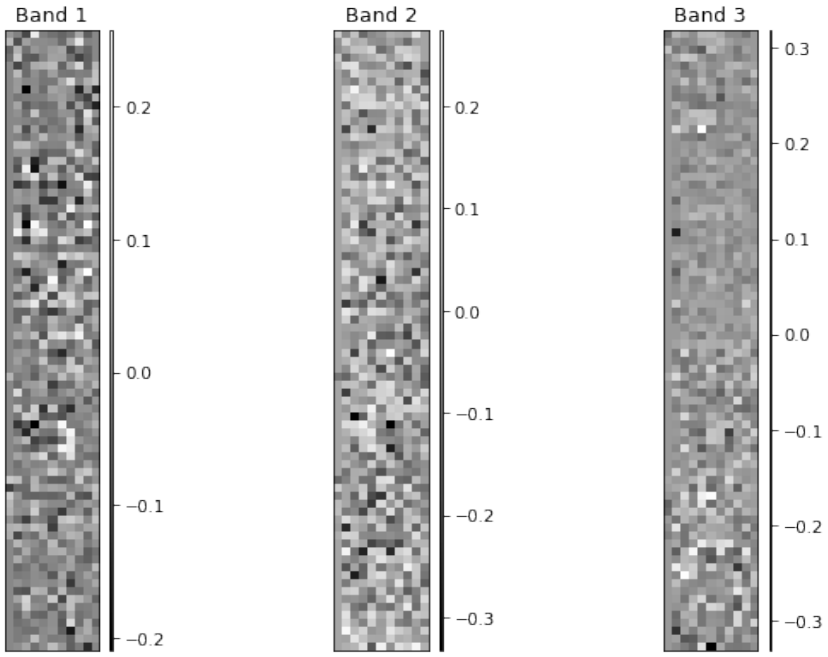
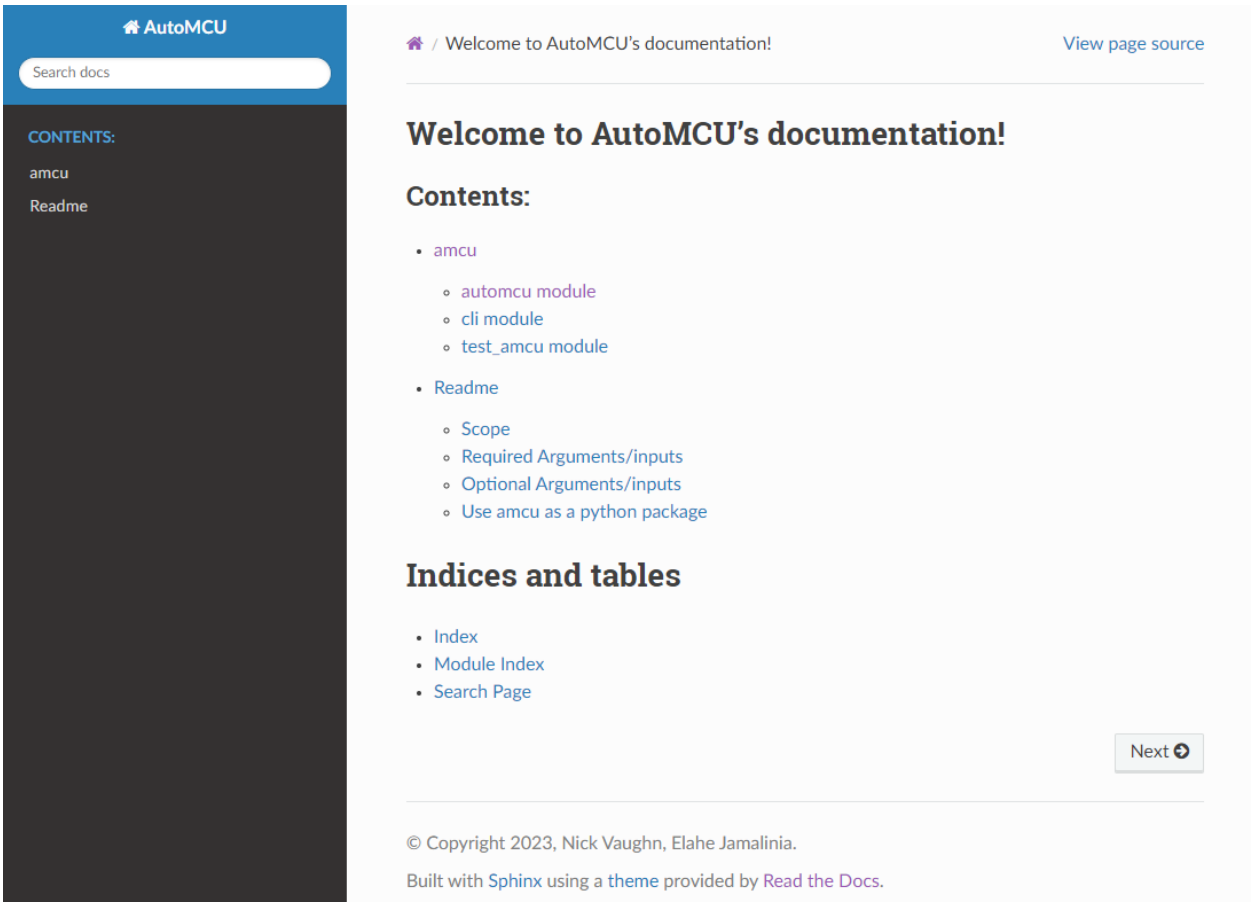


Figure 7. Difference between end member fractions between results of iterations equals to 30 and 100. Map shows the difference between a) PV fractions (band 1), b) NPV fractions (band2), c) BS fractions (band3).

5.6 Additional info

The current package comes with Readme.rst and Readme.cmd that may include other information rather than the current ATBD. Besides, Sphinx documentation in html format is provided in the package folder with the file path: .\docs\build\html\index.html. This interactive documentation enables the user to read the documentation within the code.



6. References

Asner, G., Dai, J., Hondula, K., Jamalinia, E., König, M., Martin, P., Vaughn, N., Guido, J., Shivers, S., & Duren, R. (2022). Land and Ocean Applications and Approaches for the Carbon Mapper Satellite Mission. *AGU Fall Meeting 2022*.

Asner, G. P., & Heidebrecht, K. B. (2003). Imaging spectroscopy for desertification studies: Comparing AVIRIS and EO-1 Hyperion in Argentina drylands. *IEEE Transactions on Geoscience and Remote Sensing*, *41*(6 PART I), 1283–1296. https://doi.org/10.1109/TGRS.2003.812903

Asner, G. P., & Lobell, D. B. (2000). AutoSWIR : A SPECTRAL UNMIXING ALGORITHM USING 2000-2400 nm ENDMEMBERS AND MONTE CARLO ANALYSIS. *Ecological Research*.

Asner Gregory, P., & Lobell David, B. (2000). A biogeophysical approach for automated SWIR unmizing of soils and vegetation. *Remote Sensing of Environment*, *74*(1), 99–112.
Gregory P. Asner and Kathleen B. Heidebrecht. (2000). *Spectral Unmixing of Vegetation, Soil and Dry Carbon in Arid Regions: Comparing Multispectral and Hyperspectral Observations*. *23*(19), 1–14.

Huemmrich, K. F., & Goward, S. N. (1997). Vegetation canopy PAR absorptance and NDVI: An assessment for ten tree species with the SAIL model. *Remote Sensing of Environment*, *61*(2), 254–269. https://doi.org/10.1016/S0034-4257(97)00042-4

Li, X., Zheng, G., Wang, J., Ji, C., Sun, B., and Gao, Z. (2016). Comparison of Methods for Estimating Fractional Cover of Photosynthetic and Non-Photosynthetic Vegetation in the Otindag Sandy Land Using GF-1 Wide-Field View Data. Remote Sensing 8:800.

Kim, Y., Yang, Z., Cohen, W. B., Pflugmacher, D., Lauver, C. L., & Vankat, J. L. (2009). Distinguishing between live and dead standing tree biomass on the North Rim of Grand Canyon National Park, USA using small-footprint lidar data. *Remote Sensing of Environment*, *113*(11), 2499–2510. https://doi.org/10.1016/j.rse.2009.07.010

Numata, I., Roberts, D. A., Chadwick, O. A., Schimel, J. P., Galvão, L. S., & Soares, J. v. (2008). Evaluation of hyperspectral data for pasture estimate in the Brazilian Amazon using field and imaging spectrometers. *Remote Sensing of Environment*, *112*(4), 1569–1583. https://doi.org/10.1016/j.rse.2007.08.014

Serbin, G., Raymond Hunt, E., Daughtry, C. S. T., & McCarty, G. W. (2013). Assessment of spectral indices for cover estimation of senescent vegetation. *Remote Sensing Letters*, *4*(6), 552–560. https://doi.org/10.1080/2150704X.2013.767479