



HyperGUI: A Web Application for Hyperspectral Image Analysis and Data Extraction

**SOFTWARE
METAPAPER**

SHAYAN KABIRI 

SHARON M. O'ROURKE 

*Author affiliations can be found in the back matter of this article

]u[ubiquity press

ABSTRACT

HyperGUI is an intuitive R-based web application crafted to streamline the loading of raw hyperspectral images while offering essential processing functions for exploring hyperspectral data. Users can easily load, process, and visualize benchtop or aerial hyperspectral images, with the flexibility to export them to R, Python, MATLAB, or raster data frames for further analysis. HyperGUI's source code is available on GitHub, and it is accessible online via shinyapps.io.

CORRESPONDING AUTHOR:

Shayan Kabiri

PhD student, UCD School of Biosystems & Food Engineering, University College Dublin, Belfield, Dublin, Ireland
shayan.kabiri@ucdconnect.ie

KEYWORDS:

Hyperspectral imaging;
hyperspectral image processing; web apps; R; shiny

TO CITE THIS ARTICLE:

Kabiri S, O'Rourke SM 2024
HyperGUI: A Web Application for Hyperspectral Image Analysis and Data Extraction. *Journal of Open Research Software*, 12: 9. DOI: <https://doi.org/10.5334/jors.509>

(1) OVERVIEW

INTRODUCTION

There are numerous platforms available for hyperspectral image processing, but these platforms are either proprietary and closed source such as ENVI [1] or when open-source and free, are only accessible within programming languages, such as Spectral Python [2] a package for handling hyperspectral data in Python and, hyperSpec [3], terra [4] and hsdar [5], three hyperspectral image analysis packages for R programming language. This can often present as a barrier to new users of hyperspectral data.

Web apps are quickly gaining popularity with scientists for their ability to create data analysis software that is robust to write and run, open-sourced, high-throughput and with strong visualization abilities [6–8]. ‘Shiny’ [9] is an object oriented and reactive web application development package within R programming language, that enables scientists to develop interactive graphical user interface (GUI) based applications for data analysis purposes.

While there are similar GUI apps such as MIC-Surgery-Heidelberg/HyperGui for medial hyperspectral imaging, these are highly specialized for a specific task within a certain project, not well documented or published and are not friendly to general users. Here we introduce a webapp for general hyperspectral imaging preprocessing and data conversion which is accessible and friendly to a wide variety of users.

Here a web app named HyperGUI was developed based on Shiny, to handle loading, processing, and visualizing of hyperspectral images with ease. The objectives were to develop a software with functions to (1) load benchtop or aerial hyperspectral images, (2) calculate the surface reflectance based on dark and white current in the case of benchtop samples and

based on ground spectral reference samples in the case of aerial images, (3) calculate PCA for a desired spectrum and save the reduced data based on percentage of variance explained and (4) visualize the hyperspectral image and the spectrum of ROIs for informative and sample selection purposes.

IMPLEMENTATION AND ARCHITECTURE

HyperGUI is based on R’s web application development framework package, Shiny [9]. The overall architecture of the web app is divided into a server and a user interface (UI), with the server handling the file loading/saving and calculations (Figure 1), and the UI presenting interactive graphical input/output. The source code is available at <https://github.com/shaikabiri/hyperGUI>. The application can be run either on a localhost via RStudio [10] or online via shinyapps.io at <https://urye8b-shayan-kabiri.shinyapps.io/hyperGUI/>.

To run the application on the localhost the following procedure should be followed. The below codes can be run in either R or RStudio. The video tutorials for the installation and the application usage can be found at <https://youtube.com/@hyperGUI>.

```
#install the devtools package first
install.packages('devtools')

#install the package using devtools:
devtools::install_github('https://github.com/shaikabiri/hyperGUI')

#load the package
library(hyperGUI)

#run the app by the following command
hyperGUI::run_app()
```

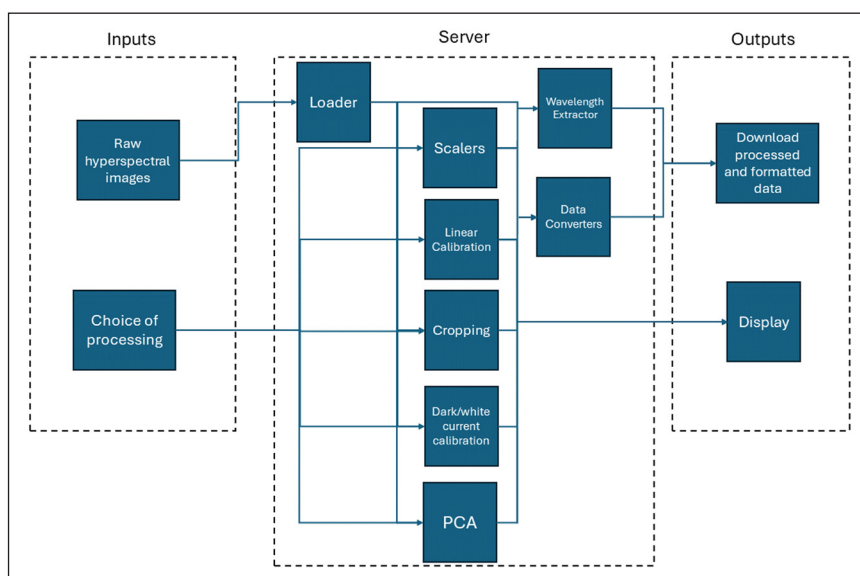


Figure 1 The architecture diagram for components of HyperGUI and their relationships.

The software's GUI is divided into a sidebar and a display area. The display area presents the hyperspectral image in form of a monochromatic image of a wavelength of a choice, pseudo-RGB image of three wavelengths of choice or a pseudo-RGB image of three most important components of a PCA. Moreover, the displayed image can be interacted with to crop the data or display spectra for regions of interest or spectra of interest in a plot below the displayed image. The sidebar controls the type of hyperspectral image, either benchtop or aerial, file load, data save and changes to parameters of the displayed image including wavelengths of interest, and mode of display (Figure 2).

Data management and processing

Decoding loaded raw image is handled using hyperSpec [3] and terra [4] packages. A custom header reader function has been written to handle variety in header file formatting. In the case of benchtop hyperspectral images, dark current and white current raw images and raw files can be loaded to calculate surface reflectance. Data can

be scaled both in each hyperplane or for each pixel along hyperplanes. For aerial hyperspectral images, ROI of the ground references in shapefile format and spectral data of ground references can be loaded to calculate surface reflectance. For aerial images, a simple linear regression for each band is performed with goodness of fit shown at the finishing of calibration for each band. In case that no reference is available for hyperspectral images, there's an option to perform a simple dark current pixel subtraction. This algorithm takes the darkest pixel in each band and subtracts the whole band with this value. The cropping feature not only crops the image but the whole hypercube. At any stage of analysis, the hypercube represented by the displayed image can be saved as a MATLAB, Python, R, or raster data frame. A tab delimited table of wavelengths can also be saved separately.

Data visualization

Plots are produced by ggplot2 [11] package and projected in GUI. A small part of the application that handles interaction with the plots is written in JavaScript and runs

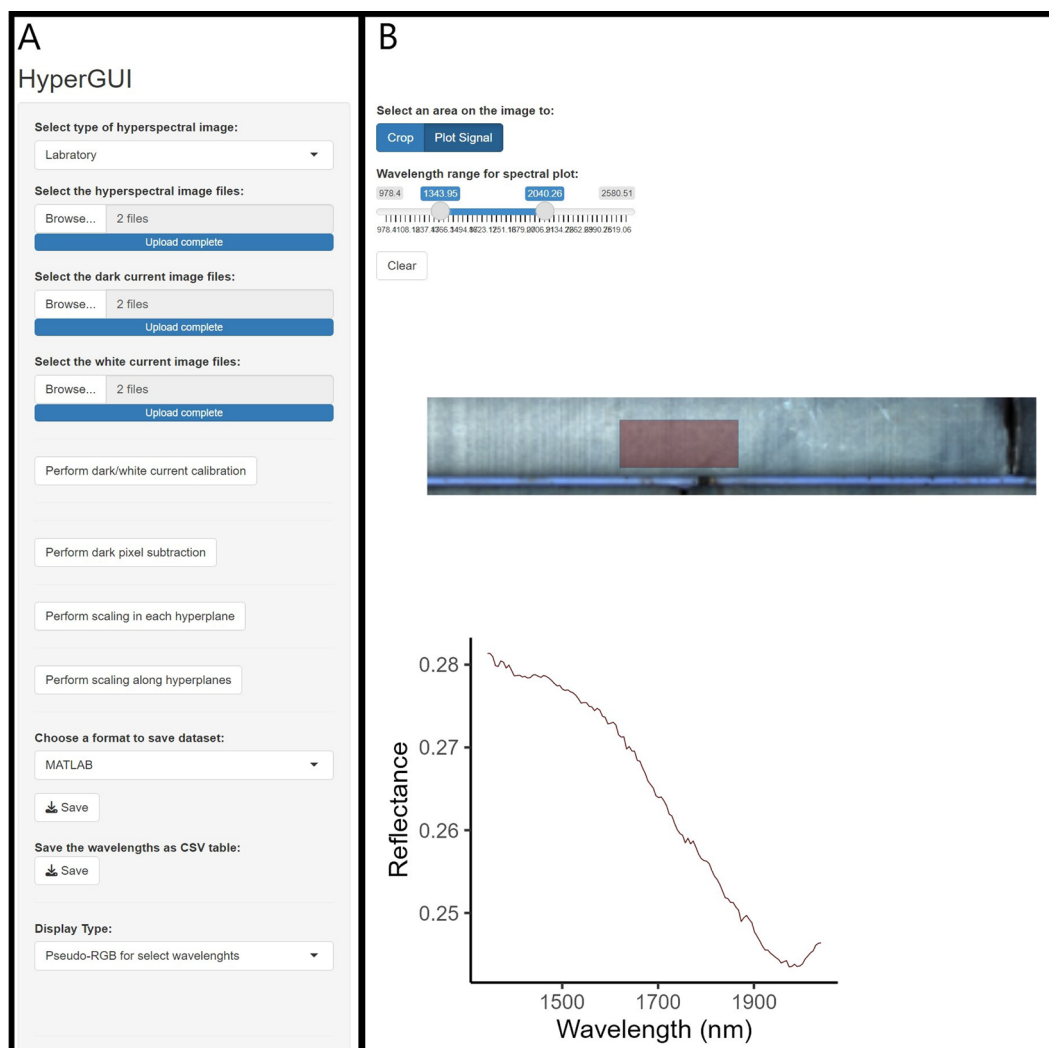


Figure 2 Organization of software's GUI. (A) The sidebar panel that controls file input/output, parameters of displayed image and mode of displayed image (B) The display panel, presenting the hyperspectral image, and interactive cropping or region of interest selection (grey rectangle) to display spectra on a separate plot below.

natively in R using shinyjs [12] package. All plots in the current article are produced only in HyperGUI.

Principal component analysis

Principal Component Analysis (PCA) can be calculated and visualized as a pseudo-RGB image for selected spectrum, and importance of each component can be displayed separately. Reduced data can be saved by selecting a proportion of variance that can be explained by Principal Components (Figure 3). PCA can be utilised here for data compression [13] or informing ROI for reference sampling.

QUALITY CONTROL

Two hyperspectral image samples from Eartha Science applications were used to test the software, an aerial image from a published study [14], and a benchtop image from the Geological Survey Ireland (GSI) hyperspectral imaging data project [15] hereafter called sample A and sample B, respectively. Sample A is an aerially acquired visible to short wave infrared (SWIR) image of Gudalur,

India, with a spectral range from 376 to 2500 nm, acquired by the AVIRIS-NG sensor. Ground references were taken using a field spectrometer (Spectra Vista Corporation, HR-1024i, USA) with a spectral range of 350 to 2500 nm. Sample B is a single core box, consisting of SWIR image of 105.98 to 110.35 m section of a sediment core from drillhole GSI-17-007 of Geological Survey Ireland's hyperspectral project, with spectral range of 980 to 2580 nm. Software was tested on a laptop PC with AMD Ryzen 7 4800H CPU and 12 GB of RAM on Windows 11 and web browser Chrome Version 124.0.6367.202 (Official Build) (64-bit).

Overall, the web app shows acceptable performance, with no process taking more than a few seconds except for linear calibration and PCA which took approximately 3 minutes to finish processing. The heaviest task was taking a PCA on uncropped sample B, which consumed approximately 4GB of memory and 27% of CPU power, other tasks were light in memory and CPU consumption. Functionality of all features were tested successfully. Figures 4 and 5 show the results of dark and white

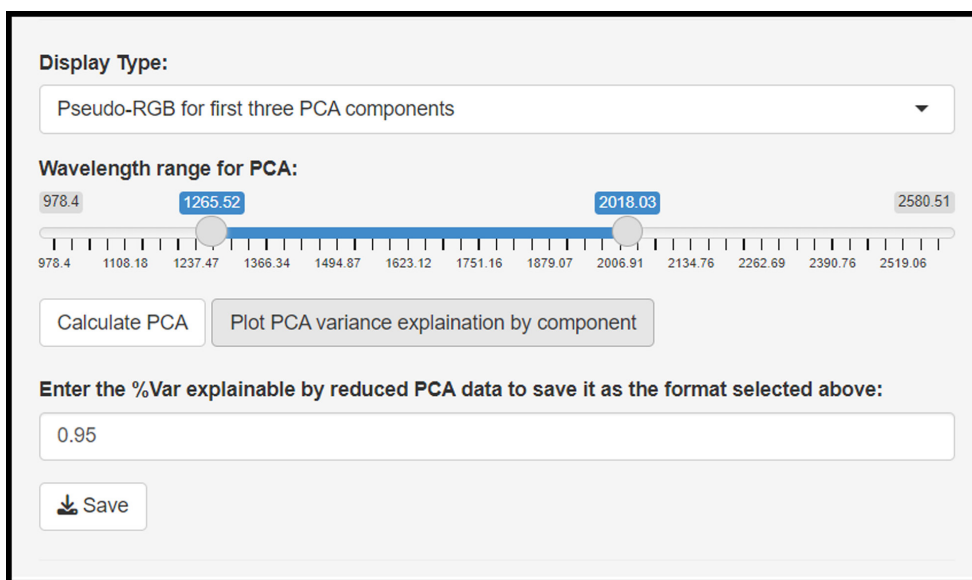


Figure 3 Options panel for PCA in sidebar section. A spectrum can be selected to perform PCA. PCA variance explanation by component can be plotted in display section. A proportion of variance to be explained by reduced data can be selected and the resulting reduced data can be saved.

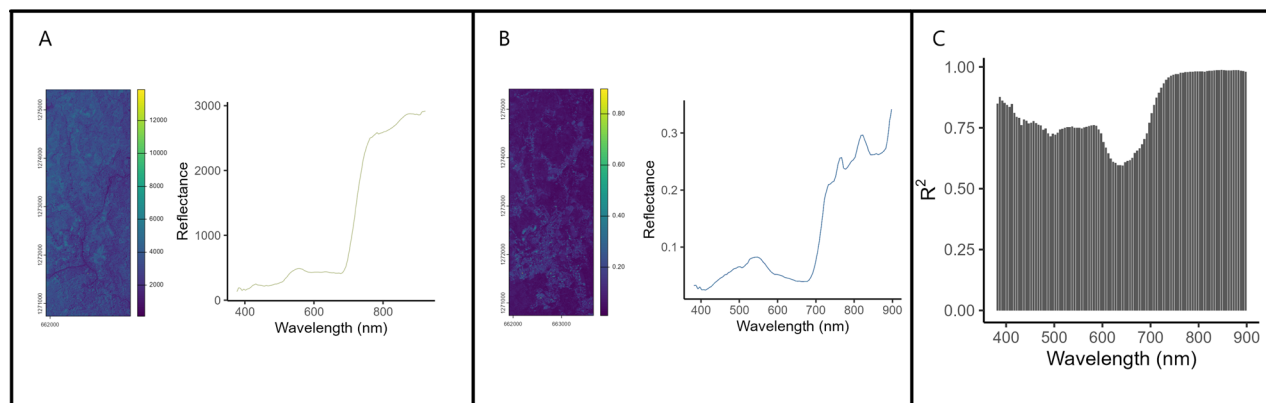


Figure 4 Spectra and the image for (A) sample A before linear calibration and (B) after linear calibration with ground references. (C) Goodness of fit of calibration for each spectral band.

current correction and linear calibration for Sample A and B. The linear calibration for sample A closely matches with calibrated spectra published by Jha and Nidamanuri [16], even though a more sophisticated sensor-specific method was used in that study. The results of dark and white current for sample B matches the results published by GSI [17] (URL: https://gsi.geodata.gov.ie/downloads/Bedrock/Data/Hyperspectral_Sample_data.zip). Figure 6 shows PCA results for sample A and B. Saved data frames were verified in MATLAB 2021b, Python 3.10, R 4.2.2 and QGIS 3.22.1.

(2) AVAILABILITY

The source code is available at <https://github.com/shaikabiri/hyperGUI>. The application can be run either on a localhost via RStudio [10] or on a standalone server. A server-based version of the web app is currently running at <https://urye8b-shayan-kabiri.shinyapps.io/hyperGUI/>, but the localhost is preferred because of large file size of hyperspectral data. A detailed user guide is available

at web app's repository with links to example data. Experts are invited to use the software, report errors, and suggest, write, or edit functions, via GitHub repository or by contacting the author.

OPERATING SYSTEM

Windows, Linux and Mac

PROGRAMMING LANGUAGE

R version 4.3.3

ADDITIONAL SYSTEM REQUIREMENTS

Minimum ram 4GB

DEPENDENCIES

RStudio 2023.12.1

SOFTWARE LOCATION

Archive (e.g. institutional repository, general repository) (required – please see instructions on journal website for depositing archive copy of software in a suitable repository)

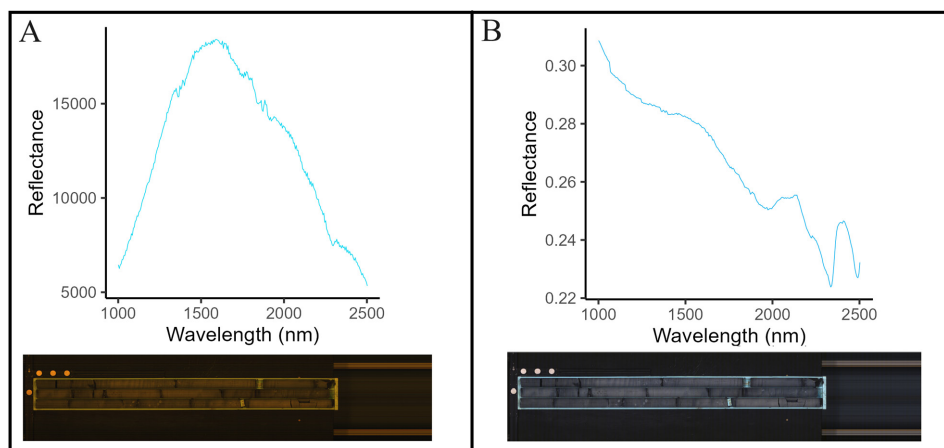


Figure 5 Spectra and pseudo-RGB image at wavelengths R = 1506, G = 1243 and B = 978.4 for sample B (A) before and (B) after dark and white current correction.

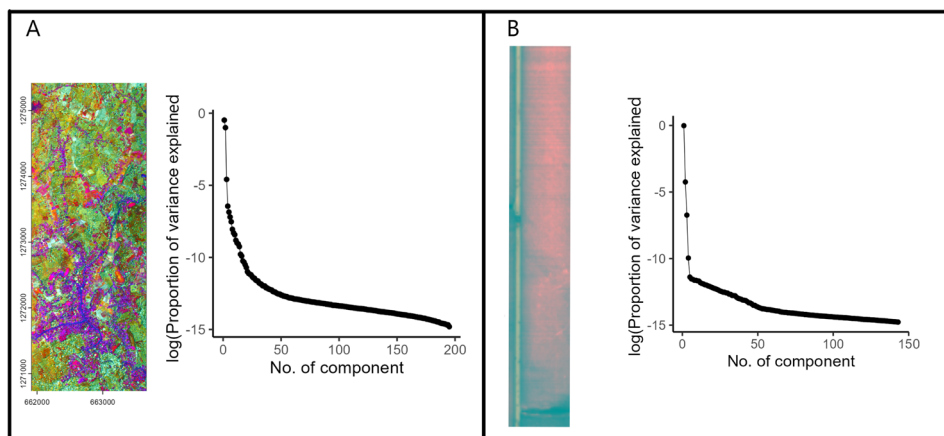


Figure 6 Resulting pseudo-RGB image and proportion of variance explained by component for (A) sample A and (B) part of a core in sample B, for spectra of 1000 to 2500 nm.

Name: HyperGUI: Webapp for Hyperspectral Image I/O

Persistent identifier: <https://doi.org/10.5281/zenodo.11356643>

Licence: MIT License

Publisher: Shayan Kabiri

Version published: 1.11

Date published: 27/5/2024

Code repository (e.g. SourceForge, GitHub etc.) (required)

Name: hyperGUI

Identifier: <https://github.com/shaikabiri/hyperGUI>

Licence: MIT License

Date published: 27/5/2024

Licence: Open license under which the software is licensed here

Date published: dd/mm/yy

LANGUAGE

English

(3) REUSE POTENTIAL

Our software has already been integrated into our research group's workflow and will be acknowledged in forthcoming publications. We've noticed numerous instances where individuals encounter challenges when attempting to convert raw hyperspectral images into conventional data frames using R, Python, and MATLAB. By releasing this software, we aim to provide other researchers with a user-friendly web application tailored for this task, ensuring accessibility, and facilitating repeatability, which also guarantees reusability and citability of the software.

FUNDING INFORMATION

This work was supported by the Irish Research Council [grant number IRCLA/2017/137].

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR AFFILIATIONS

Shayan Kabiri  orcid.org/0000-0003-3231-5843

PhD student, UCD School of Biosystems & Food Engineering, University College Dublin, Belfield, Dublin, Ireland

Sharon M. O'Rourke  orcid.org/0000-0002-4565-1689

Assistant Professor, UCD School of Biosystems & Food Engineering, University College Dublin, Belfield, Dublin, Ireland

REFERENCES

- Solutions HG.** ENVI 5.7. Broomfield, CO, United States: Harris Geospatial Solutions; 2023.
- Boggs T.** Spectral Python. 0.21 ed; 2020.
- Beleites C, Sergio V.** hyperSpec: a package to handle hyperspectral data sets in R; 2023.
- Hijmans RJ.** terra: Spatial Data Analysis; 2023.
- Lehnert LW, Meyer H, Obermeier WA,** et al. Hyperspectral Data Analysis in R: The hsdar Package. *Journal of Statistical Software*. 2019; 89: 1–23. DOI: <https://doi.org/10.18637/jss.v089.i12>
- Abriata LA.** Web Apps Come of Age for Molecular Sciences. *Informatics*. 2017; 4: 28. DOI: <https://doi.org/10.3390/informatics4030028>
- Goedhart J.** PlotTwist: A web app for plotting and annotating continuous data. *PLOS Biology*. 2020; 18: e3000581. DOI: <https://doi.org/10.1371/journal.pbio.3000581>
- Goedhart J, Luijsterburg MS.** VolcaNoseR is a web app for creating, exploring, labeling and sharing volcano plots. *Scientific Reports*. 2020; 10: 20560. DOI: <https://doi.org/10.1038/s41598-020-76603-3>
- Chang W, Cheng J, Allaire JJ,** et al. shiny: Web Application Framework for R; 2023.
- Posit t.** RStudio: Integrated Development Environment for R; 2022.
- Wickham H.** ggplot2: Elegant Graphics for Data Analysis; 2016. DOI: https://doi.org/10.1007/978-3-319-24277-4_9
- Attali D.** shinyjs: Easily Improve the User Experience of Your Shiny Apps in Seconds; 2022.
- Ren J, Zabalza J, Marshall S,** et al. Effective Feature Extraction and Data Reduction in Remote Sensing Using Hyperspectral Imaging [Applications Corner]. *IEEE Signal Processing Magazine*. 2014; 31: 149–154. DOI: <https://doi.org/10.1109/MSP.2014.2312071>
- Jha SS, Kumar M and Nidamanuri RR.** Multi-platform optical remote sensing dataset for target detection. *Data in Brief*. 2020; 33: 106362. DOI: <https://doi.org/10.1016/j.dib.2020.106362>
- Irish Public Sector Data I.** Hyperspectral sample data (GSI-17-007, box 25); 2023.
- Jha S and Nidamanuri R.** Gudalur Spectral Target Detection (GST-D): ANew Benchmark Dataset and Engineered Material Target Detection in Multi-Platform Remote Sensing Data. *Remote Sensing*. 2020; 12: 2145. DOI: <https://doi.org/10.3390/rs12132145>
- GSI GSI.** Bedrock. In: Ireland GS, (ed.); 2019.

TO CITE THIS ARTICLE:

Kabiri S, O'Rourke SM 2024 HyperGUI: A Web Application for Hyperspectral Image Analysis and Data Extraction. *Journal of Open Research Software*, 12: 9. DOI: <https://doi.org/10.5334/jors.509>

Submitted: 30 March 2024 **Accepted:** 31 May 2024 **Published:** 18 July 2024

COPYRIGHT:

© 2024 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

Journal of Open Research Software is a peer-reviewed open access journal published by Ubiquity Press.

