Geodata-Harvester: A Python package to jumpstart geospatial data extraction and analysis

Sebastian Haan, Januar Harianto, Nathaniel Butterworth, Thomas Bishop

21 June 2023

Summary

`Geodata-Harvester` is a user-friendly Python package that enables researchers with reusable workflows and software tools for automatic extraction, processing and analysis of geo-spatial and environmental data. User provided data is auto-completed with a suitable set of spatial- and temporal-aligned covariates as a ready-made dataset for machine learning models. All data layer maps are automatically extracted and aligned for a specific region and time period.

The `Geodata-Harvester` is designed to be modular and extensible, offering multiple front-end notebooks and use case scenarios to encourage interaction and experimentation with the pipeline. With its connectivity support to the Google Earth Engine (GEE) API (Gorelick et al. 2017) and integrating the latest GEE add-ons (Wu 2020; Montero 2021; David Montero 2022), the software also enables users to perform petabyte-scale operations, including temporal cloud/shadow masking and automatic calculation of spectral indices.

Statement of Need

There is an enormous amount of national/global space-time data that is free and accessible, such as numerous satellite platforms, weather, terrain, soil, and landscape data. Currently, a researcher must search through several places for these resources. This includes publication search engines, specialist aggregators or repositories, R/Python libraries, between statistical packages, GitHub, on the web and through personal contacts. Many data layers require a number of post-processing steps that a user can undertake to extract meaning, e.g. spatial alignment, temporal means, aggregating in time. The data are then able to be selected and extracted in the desired format, and stored to either their local desktop, or virtual desktop with access to a high compute workspace. All of the above is a non-trivial task and the ideal experience for researchers would be to be able to find and extract key foundational datasets (such as climate, landscape, soil, and remote sensed data) at once given the required spatial, area and temporal range for their analysis.

The need for a `Geodata-Harvester` emerges from the increasing demand for an extendable, automated, and reusable system for geo-spatial and environmental data extraction and machine learning model preparation. The `Geodata-Harvester` software allows researchers to jumpstart their analysis with a ready-made set of spatial-temporal aligned raster maps and dataframes. Unlike geodata-handler packages such as `osgeo` libraries, `rasterio`[1], `rioxarray`[2], `pystack`[3], `intake` plugins[4], the Geodata-Harvester builds on top of these resources a cohesive workflow for automatic data extraction from multiple geospatial sources at once. Its unique features include reproducible workflows via YAML settings files, connectivity to a wide range of geodata APIs, automatic data retrieval and processing, and high-level integration of Google Earth Engine capabilities. The aim of this on-going project is to offer a flexible all-in-one solution, enabling efficient geospatial research and machine learning applications.

[1] https://corteva.github.io/rioxarray/stable/
[2] https://rasterio.readthedocs.io/en/latest/
[3] https://pystac.readthedocs.io/en/stable/
[4] https://intake.readthedocs.io/en/latest/

Tutorials and Workshops

To get started, some example workflows and tutorials are provided as:

- Jupyter notebooks
- Geodata-Harvester workshop material.
- Geodata-Harvester documentation
- Settings_Overview
- GEE harvester project: eeharvest
- R-package wrapper: dataharvesteR

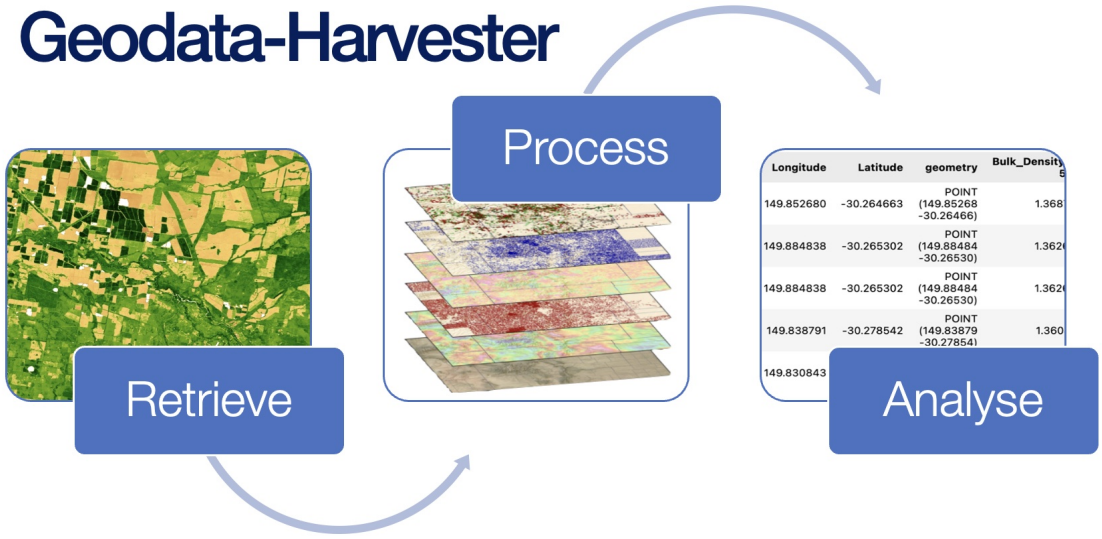

Figure 1: `Geodata-Harvester` overview

Functionality and Key Features

The main goal of the Data Harvester is to enable researchers with reusable workflows for automatic data extraction and processing:

1. Retrieve: given set of locations, automatically access and download multiple data sources (APIs) from a diverse range of geospatial and soil data sources
2. Process: Spatial and temporal processing, conversion to dataframes and custom raster-files
3. Output: Ready-made dataset for machine learning (training set and prediction mapping)

Below is a list of main features available for the `Geodata-Harvester` package. Please check the project Github webpage and notebooks for examples, data selection, and other settings.

- enabling reproducible workflows via YAML settings files
- automatic data retrieval from geodata APIs for given locations and dates
- automatic download and spatial-temporal processing of geo-spatial maps for user-specified bounding box, resolution, and time-scale
- support for multiple temporal aggregation options and spatial-temporal buffer
- automatic extraction of retrieved data into ready-made dataframes for ML training
- automatic generation of ready-made aligned maps and data for ML prediction models
- visualisation of downloaded and aligned maps
- support for saving and loading settings via interactive widgets
- with connectivity support to the Google Earth Engine API, perform petabyte-scale operations which include temporal cloud/shadow masking and automatic calculation of spectral indices
- easy install via conda-forge or PyPI package index

Data Sources

The following data sources are currently implemented:

- Soil and Landscape Grid of Australia (SLGA)
- SILO Climate Database, Australia (Jeffrey et al. 2001)
- National Digital Elevation Model (DEM) 1 Second Hydrologically Enforced, Australia
- Digital Earth Australia (DEA) Geoscience Earth Observations, Australia (Krause et al. 2021)
- GSKY Data Server for DEA Geoscience Earth Observations, Australia
- Radiometric Data, Australia
- Google Earth Engine Data (GEE account needed)

A detailed list of all available layers and their description can be found in Data Overview. The `Geodata-Harvester` is designed to be extendable and new data source modules can be added (see adding new data source guidelines).

Acknowledgements

This software was developed by the Sydney Informatics Hub, a core research facility of the University of Sydney, as part of the Geodata Harvester project for the Agricultural Research Federation (AgReFed). If you make use of this software for your research project, please cite this paper or include the following acknowledgment:

"This research was supported by the Sydney Informatics Hub, a Core Research Facility of the University of Sydney, and the Agricultural Research Federation (AgReFed)."

AgReFed is supported by the Australian Research Data Commons (ARDC) and the Australian Government through the National Collaborative Research Infrastructure Strategy (NCRIS).

References

David Montero, Miguel D. Mahecha, Cesar Aybar. 2022. "Spectral: Awesome Spectral Indices Deployed via the Google Earth Engine JavaScript API." *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLVIII-4/W1-2022: 301–6. https://doi.org/10.5194/isprs-archives-XLVIII-4-W1-2022-301-2022.

Gorelick, Noel, Matt Hancher, Mike Dixon, Simon Ilyushchenko, David Thau, and Rebecca Moore. 2017. "Google Earth Engine: Planetary-Scale Geospatial Analysis for Everyone." *Remote Sensing of Environment* 202: 18–27. https://doi.org/10.1016/j.rse.2017.06.031.

Jeffrey, Stephen J, John O Carter, Keith B Moodie, and Alan R Beswick. 2001. "Using Spatial Interpolation to Construct a Comprehensive Archive of Australian Climate Data." *Environmental Modelling & Software* 16 (4): 309–30. https://doi.org/10.1016/S1364-8152(01)00008-1.

Krause, C, B Dunn, R Bishop-Taylor, C Adams, C Burton, M Alger, S Chua, et al. 2021. "Digital Earth Australia Notebooks and Tools Repository." *Geoscience Australia*.

Montero, David. 2021. "Eemont: A Python Package That Extends Google Earth Engine." *Journal of Open Source Software* 6 (62): 3168. https://doi.org/10.21105/joss.03168.

Wu, Qiusheng. 2020. "Geemap: A Python Package for Interactive Mapping with Google Earth Engine." *Journal of Open Source Software* 5 (51): 2305. https://doi.org/10.21105/joss.02305.