geemap: A Python package for interactive mapping with Google Earth Engine

Qiusheng Wu

1 Department of Geography, University of Tennessee, Knoxville, TN 37996, United States

Summary

geremap is a Python package for interactive mapping with Google Earth Engine (GEE), which is a cloud computing platform with a multi-petabyte catalog of satellite imagery and geospatial datasets (e.g., Landsat, Sentinel, MODIS, NAIP) (Gorelick et al., 2017). During the past few years, GEE has become very popular in the geospatial community and it has empowered numerous environmental applications at local, regional, and global scales. Some of the notable environmental applications include mapping global forest change (Hansen et al., 2013), global urban change (Liu et al., 2020), global surface water change (Pekel, Cottam, Gorelick, & Belward, 2016), wetland inundation dynamics (Wu et al., 2019), vegetation phenology (Li et al., 2019), and time series analysis (Kennedy et al., 2018).

GEE provides both JavaScript and Python APIs for making computational requests to the Earth Engine servers. Compared with the comprehensive documentation and interactive IDE (i.e., GEE JavaScript Code Editor) of the GEE JavaScript API, the GEE Python API lacks good documentation and lacks functionality for visualizing results interactively. The geemap Python package is created to fill this gap. It is built upon ipyleaflet and ipywidgets, enabling GEE users to analyze and visualize Earth Engine datasets interactively with Jupyter notebooks.

geemap Audience

geemap is intended for students and researchers who would like to utilize the Python ecosystem of diverse libraries and tools to explore Google Earth Engine. It is also designed for existing GEE users who would like to transition from the GEE JavaScript API to a Python API. The automated JavaScript-to-Python conversion module of the geemap package can greatly reduce the time needed to convert existing GEE JavaScripts to Python scripts and Jupyter notebooks.

geemap Functionality

The interactive mapping functionality of the geemap package is built upon ipyleaflet and folium, both of which rely on Jupyter notebooks for creating interactive maps. A key difference between ipyleaflet and folium is that ipyleaflet is built upon ipywidgets and allows bidirectional communication between the frontend and the backend, enabling the use of the map to capture user input, while folium is meant for displaying static data only (QuantStack, 2019). It should be noted that Google Colab currently does not support ipyleaflet. Therefore, if one wants to use geemap on Google Colab, one should import geemap.eefolium as geemap, which provides limited interactive mapping functionality. To utilize the full interactive mapping...
functionality of `geemap`, one should import `geemap` on a local computer or secured server with Jupyter notebook installed.

The key functionality of `geemap` is organized into several modules:

- **geemap**: the main module for interactive mapping with Google Earth Engine, ipyleaflet, and ipywidgets.
- **eefolium**: a module for interactive mapping with Earth Engine and folium. It is designed for users to run geemap with Google Colab.
- **conversion**: utilities for automatically converting Earth Engine JavaScripts to Python scripts and Jupyter notebooks.
- **basemaps**: a module for adding various XYZ and WMS tiled basemaps.
- **legends**: a module for adding customized legends to interactive maps.

**geemap Tutorials**

Various tutorials and documentation are available for using `geemap`, including:

- 20+ video tutorials with corresponding notebook examples
- 360+ Jupyter notebook examples for using Google Earth Engine
- Complete documentation on geemap modules and methods

**Acknowledgements**

The author would like to thank the developers of ipyleaflet and ipywidgets, which empower the interactive mapping functionality of `geemap`, including Martin Renou, Sylvain Corlay, and David Brochart. The author would also like to acknowledge source code contributions from Justin Braaten, Cesar Aybar, Oliver Burdekin, Diego Garcia Diaz, and Stephan Büttig.

**References**


