Nancy Fisher

**How to Guide**

**Introduction**

In the field of Cultural Resource Management (CRM), it is crucial for clients to understand the project area before work begins. This understanding is typically achieved through a scoping report, which provides detailed insights into the environment of the project area. The code in this guide automates and streamlines the creation of geology maps, a critical component of these scoping reports.

A geology map provides essential information about the geological features in the project area, such as the type of bedrock (the information is normally based on the available data). This information is vital for assessing the feasibility and potential impact of the project, as well as identifying any geological considerations that may affect the project’s execution.

To use the map code, users must first convert KMZ files into shapefiles. This conversion process was demonstrated using ArcPro, but it can also be performed using QGIS or with a Python script. The KMZ and resulting shapefile use the projected coordinate system WGS 1984 UTM Zone 16N.

For demonstration purposes, the example code uses a mock project area centered at coordinates -86.137945, 39.882478 in Indianapolis, IN. The geology data is sourced from the USGS Geology, Geophysics, and Geochemistry Science Center. These datasets provide geologic data for the United States and Australia using a consistent rock classification, available in GIS-ready format, and are used for prospectivity modeling of basin-hosted Pb-Zn mineralization (McCafferty, A. E., et al., 2023). For the basemap, OpenStreetMap provides a detailed and up-to-date geographic context (OpenStreetMap).

This code is customizable, allowing users to input their own geology data, background data, and project area shapefiles to meet specific project needs. By focusing on automating the geology map generation and requiring pre-conversion of KMZ files, the code streamlines the production of scoping reports with minimal effort, ensuring consistency and accuracy for effective decision-making in CRM projects.

**Setup/Installation**

***Installation Instructions***

1. Clone the Repository:
2. Install Required Dependencies: Ensure you have Python installed (version 3.8 or higher). Install the required libraries using pip:

pip install matplotlib

pip install matplotlib-scalebar

pip install geopandas

pip install shapely

pip install pandas

pip install numpy

pip install contextily

pip install rasterio

***Main Dependencies***

1. matplotlib: This is the core library for creating the map and all its visual elements.
2. geopandas: Used for loading, manipulating, and plotting geospatial data (shapefiles).
3. shapely: Used for creating the bounding box for clipping operations.
4. contextily: Provides the OpenStreetMap basemap.

***Additional Dependencies***

1. matplotlib\_scalebar: Adds the scale bar to the map.
2. numpy: Used for some numerical operations, particularly in creating the color map.
3. os: Used for file path operations when saving the map.

The last three imports, patches, Line2D, and pandas, are used for specific elements like the north arrow, legend creation, and potentially some data manipulation, but they're less central to the core functionality of the script.

***Repository Link***

The code for this project is available on GitHub: https://github.com/fishernl26/my\_project

***Test Data Instructions***

The example code uses test data to demonstrate its functionality. This test data includes:

Geology Data: Sourced from the USGS Geology, Geophysics, and Geochemistry Science Center. This data can be downloaded from this link: https://www.sciencebase.gov/catalog/item/623a013ed34e915b67cddcfa

KMZ Project Area File: Sample KMZ files can be obtained from this link:

https://drive.google.com/file/d/1psTIluuJ\_93NtTFWUjUD7j6D\_yJYxYh2/view?usp=sharing

Ensure that the test data is placed in the appropriate directories as specified in the code.

**Methods**

***Overview***

This section describes the methodology used to generate a detailed geological map of a specified project area using Python. The process involves data preparation, clipping, and visualization, culminating in a map output.

***Data Preparation***

The process begins with loading the necessary data files using the geopandas library. The project area shapefile and geology shapefiles were accessed. To ensure consistent spatial reference and accurate map alignment, both datasets are transformed to the WGS 1984 UTM Zone 16N coordinate system (EPSG:32616). This transformation is crucial for maintaining spatial accuracy throughout the analysis.

***Clipping and Simplification***

The map extent is defined by calculating the central coordinates of the project area and setting a bounding box for clipping. The dimensions of this bounding box are tailored to a map scale of 1:10,000, resulting in a map size of 24 by 36 inches. Subsequently, the geology data is clipped to this bounding box using the gpd.clip function, isolating the relevant geological information for the project area. The geometry of the clipped geology shapefile is simplified with a tolerance of 100 meters. If the clipping results in an empty dataset, an alert is issued to indicate a potential lack of overlap between the geology data and the project area.

***Visualization***

The visualization process begins by creating a map with matplotlib, set to dimensions of 24 by 36 inches and a resolution of 100 DPI. Geological units are visualized using a color map from matplotlib, with each unit assigned a distinct color. OpenStreetMap is used as the basemap to provide geographic context, integrated through the contextily library. The map includes annotations such as a north arrow, added using matplotlib.patches.Arrow, and a scale bar to offer distance reference. A legend is created to differentiate between geological units and demonstrate the project area, and a title is added to provide context. The legend, positioned in the lower left corner, features color-coded geological units.

***Layout and Saving***

The map layout is adjusted to include the title, and resizing is performed if necessary to ensure the map does not exceed maximum image dimensions. Finally, the map is saved as a PNG file in the specified output directory. A confirmation message is displayed upon successful map creation and saving.

**Results**

The geological map is saved as 'geological\_map\_project\_area.png' in the designated output folder, and a confirmation message is provided to indicate successful completion. The map should appear similar, if not identical to Figure 1.

****

*Figure 1. Expected Geological Map from code*

**Troubleshooting Tips**

1. Check Folder Paths and Data: Most common issues often arise from incorrect file paths or data inconsistencies. Make sure that all file paths are accurate and accessible. Consider using absolute paths if problems persist. Additionally, verify that shapefiles are correct and contain the expected data. This can be done by examining the contents of the geodataframes after loading:

print(project\_area.head())

print(geology\_clipped.head())

1. Also make sure that the coordinate reference systems are correctly applied (use your shapefile names):

print(project\_area.crs)

print(geology\_clipped.crs)

1. Empty/Unexpected Geometries: Operations such as clipping may yield empty datasets/shapefiles or unexpected geometries. Implement checks to handle these scenarios.

if geology\_clipped.empty:

print("Warning: No geology data within the project area.")

# Consider expanding the area or using a different dataset

1. Also, verify that geometries are valid and of the correct type

print(geology\_clipped.geometry.type.value\_counts())

1. Acutal Map Errors: If the map appears incorrect or incomplete, troubleshoot by isolating and analyzing individual component. Start by plotting elements separately to identify potential issues, such as first the north arrow, then the title, etc.
   1. Verify the different elements with print statements

print(f"Map dimensions: {map\_width\_meters} x {map\_height\_meters} meters")

print(f"Number of unique geology units: {len(unique\_units)}")

**References**

McCafferty, A. E., et al. *National-Scale Geophysical, Geologic, and Mineral Resource Data and Grids for the United States, Canada, and Australia: Data in Support of the Tri-National Critical Minerals Mapping Initiative*. U.S. Geological Survey, 2023, <https://doi.org/10.5066/P970GDD5>.

OpenStreetMap Contributors. *OpenStreetMap*. OpenStreetMap, [www.openstreetmap.org](http://www.openstreetmap.org).