Fiona, Shapely, Rasterio, and PyProj: The Alternative GDAL Stack

Fiona, Shapely, Rasterio, and PyProj: The Alternative GDAL Stack

Overview

1. The Concept
2. Fiona
3. Shapely
4. Rasterio
5. PyProj

The Concept

The Concept

- GDAL is hard to use and not pythonic
  - So many strange problems because the python binding are automatically generated from C++
  - What if one were to wrap the GDAL libraries manually, ensuring a pythonic interface and native data types?

The Concept

- Enter Fiona, Shapely, Rasterio, and PyProj
    - Python packages built from the ground up to provide a pythonic and easy(er) way to use GDAL/proj4
    - Include python-specific features and convenience methods
    - Handle all the strange, difficult "features" of GDAL for the user: no more losing references!

Fiona

Fiona

- Fiona is for eature nput/ utput
  - The alternative to OGR for reading vector data
- Very lightweight
- Uses a file-like Collection for reading a data file
  - Same data source -> layer -> feature hierarchy as OGR
  - Collections are opened from a data source
  - Features are in the Collection
  - The Collection holds the "layer definition"
    - A dictionary-like mapping of field names and types
- Features are dictionary-like mappings in a format extremely similar, if not the same, as GeoJSON
- Also includes a command line utility fio

Fiona

- Some notes:
  - Requires GDAL libraries be installed to the system
    - Probably some problem with this is why I cannot get it working on my Windows machine
    - In any case, get GDAL for windows via OSGeo4W
  - Does not have any functions to manipulate geometries
    - Geometries are just a collection of points
    - Thus Shapely...

Shapely

Shapely

- Shapely is a python interface to GEOS (the underlying library powering OGR geometries)
- Geometry objects in Shapely are python objects, not C++ objects
    - Otherwise, given that they have the same underlying library the Shapely data model and operations are essentially unchanged from those of OGR
- Like Fiona, development headed by Sean Gillies of MapBox

Rasterio

Rasterio

- Rasterio is for nput/ utput
- As Fiona is to OGR, Rasterio is to GDAL
  - Raster processing in python using an abstracted version of the GDAL data model with python idioms and data types
    - Datasets are opened as a file-like object
    - Bands do not exist per se: data is read directly into an array
- Has superpowers like async processing and concurrent support (in python 3)
  - (But I believe it can release the GIL even in python 2.7, to allow multiple threads or processes to access data concurrently)
- Like Fiona, has a command line toolset called rio
- Still in its infancy: current version is 0.18.0, so a long way to go before the architecture is solidified and the docs are comprehensive
- Again, by Sean Gillies

PyProj

PyProj

- PyProj is a python-based wrapper around the proj.4 C++ library
- PyProj is by Sean Gillies, but Jeffrey Whitaker of NOAA
  - Thus it has a different feel to it than the others in this stack
  - Still, it has a very simple interface
- Again, I cannot help you if you need to do a specific transformation
  - While proj.4 does support defining custom transformations, a bit of digging through the PyProj docs turned up little to suggest PyProj support of this functionality

Bonus: GeoPandas

GeoPandas

- GeoPandas is based on the pandas tool from the scipy stack
    - (this is the python data analysis library, which aims to be a comprehensive suite of data analysis and modeling tools like R, but better because python)
- Oriented towards analyses that require data from multiple data sources that would typically be performed using a spatial database, e.g., PostGIS
- I cannot say much more, except keep an eye on this one; it is starting to generate some buzz