Interactive Computing and Processing of NASA Land Surface Observations using Google Earth Engine

Andrew Molthan$^1$, Jason Burks$^1$, and Jordan Bell$^2$

$^1$Earth Science Office / NASA Marshall Space Flight Center, Huntsville, AL
$^2$University of Alabama in Huntsville, Huntsville, Alabama

andrew.molthan@nasa.gov

Presentation 8.2, 32nd Conference on Environmental Information Processing Technologies (EIPT)
2016 (96th) AMS Annual Meeting, New Orleans, LA
Background

• Google’s Earth Engine offers a “big data” approach to processing large volumes of NASA and other remote sensing products. [https://earthengine.google.com/](https://earthengine.google.com/)

• Interfaces include a Javascript or Python-based API, useful for accessing and processing over large periods of record for Landsat and MODIS observations.
  • Other data sets are frequently added, including weather and climate model data sets, etc.

• Demonstrations here focus on exploratory efforts to perform land surface change detection related to severe weather, and other disaster events.
Javascript API (http://ee-api.appspot.com)

Online, shareable repository of code and documentation

Interface for writing Javascript routines executing data set processing

Output console for reporting text, graphs, and other outputs...

Classic “Hello World!” example...
Javascript API (http://ee-api.appspot.com)

Simple Example:
1. Retrieve Aqua MODIS NDWI for all scenes in Summer (JJA) 2010.
2. Confirm retrieval by printing information to the console.
Adding:
1. Get the same product and period for 2011.
2. Retain the maximum.
3. Difference 2011-2010
4. Display
5. Compare layers

```
// Create a 'collection', in this case Aqua MODIS Normalized Difference Water Index (NDWI)
var collection = ee.ImageCollection("MODIS/006/MOD39A3_GM_ANTARES");
// Filter that collection to the period of June 1, 2010 through August 31, 2010
var aqua_ndwi_2010 = collection.filterDate('2010-06-01', '2010-08-31');
// Create another period for Summer 2011.
var aqua_ndwi_2011 = collection.filterDate('2011-06-01', '2011-08-31');
// Create a single image to display -- the maximum for each period.
var aqua_ndwi_2010_max = aqua_ndwi_2010.max();
var aqua_ndwi_2011_max = aqua_ndwi_2011.max();
// Difference those parameters
var diff = aqua_ndwi_2011_max.subtract(aqua_ndwi_2010_max);
// Display on a map.
var ndwiVis = (min: 0, max: 1); // Grayscale visualization for NDWI
Map.addLayer(aqua_ndwi_2011_max, ndwiVis, 'Aqua MODIS Max NDWI Summer 2011');
Map.addLayer(aqua_ndwi_2010_max, ndwiVis, 'Aqua MODIS Max NDWI Summer 2010');
Map.addLayer(diff, diffVis, 'Difference NDWI Max');
centerMap(-95, 40); // Center display on the Nebraska/Iowa/Missouri borders
```
Adding:
1. Get the same product and period for 2011.
2. Retain the maximum.
3. Difference 2011-2010
4. Display
5. Compare layers
Python API (https://developers.google.com/earth-engine/python_install?hl=en)

Python packages are available for the API as well.

Allows users to authenticate their interaction with Earth Engine and write scripts, etc. to perform processing and other Python manipulations.

Preferable in some ways to using a web client for interaction and coding.

---

Python installation

The Earth Engine Python API is distributed as a Python package that is hosted on Github. The following instructions give an overview of installing the Google Earth Engine Python API. To use the Earth Engine Python API you’ll need to install the client library and its dependencies on your computer and then set up authentication credentials.

Installing the client library

Ubuntu Linux & Mac OS X installation

After the initial set up, the installation flows for Mac OS X and Ubuntu are nearly identical.

1. Set up pip and Python

PIP is a package manager for Python. The following installation instructions assume that you are using it.

Ubuntu Linux

Verify that you have Python 2.6 or 2.7:

```python
python --version
```

If needed, install 2.6 or 2.7 with `apt-get`. Then pip can be installed with:

```bash
sudo apt-get install python-pip
```

Mac OS X

The installation instructions assume that you are using Mac OS X 10.9+, the Homebrew Mac OS package manager, and the pip Python package manager. Feel free to use a different package manager such as Fink or MacPorts if you prefer.
Applications: Land Surface Change

Change in Maximum NDVI (May): 2011 versus 2010

Cordova (EF4)

Tuscaloosa-Birmingham (EF4)

Shoal Creek (EF4)
Applications: Land Surface Change

Change in Maximum NDVI (May): 2012 versus 2010
Applications: Land Surface Change

Change in Maximum NDVI (May): 2013 versus 2010

Cordova (EF4)

Shoal Creek (EF4)

Tuscaloosa-Birmingham (EF4)
Applications: Hail Damage

BYO-Data: Previously, Google Maps Engine allowed you to import your own raster data for use with other Engine-provided data sets. Recent updates appear to use a new ‘Assets’ management feature.

NSSL MESH (Hail Size) Imported to Earth Engine

August 18, 2011
Applications: Hail Damage

BYO-Data: Imported data sets can be used alongside other Earth Engine holdings to provide additional information for algorithms, or Earth Engine techniques can be applied to input data sets.

Example: Earth Engine MODIS imagery used to analyze vegetation: surface reflectance, vegetation index, temperature. Identify anomalies, restrict by size, shape, orientation, and hail size.

Based upon technique explored by Bell (2015), discussed on Thursday, 2:30, rm 252
Strengths / Weaknesses

• Strengths
  • Earth Engine data holdings are tremendous, and the streamlined Javascript/Python API is capable of processing large volumes of data very quickly.
  • Relatively simple, big data platform for exploring various remote sensing products and capabilities.
  • Numerous and increasing number of high volume data sets available that could be difficult to download and process locally.

• Weaknesses
  • Primarily focused on raster types of data – large gridded or swath products of comparable resolution.
  • Most products still require some understanding of the raw data – many appear to be a literal depositing of the source data with no additional post-processing.
    • Ex: Temperatures in scaled units, not K
  • Documentation could benefit from a few use examples, and some experiences where documentation or errors were cryptic.
Questions?

• Check out some related talks!
  • Near-Real Time Severe Weather Damage Identification Algorithm for Vegetation: Development and Early Results
    • J. Bell, University of Alabama in Huntsville
      • Thursday, 3:30 pm, 252

• andrew.molthan@nasa.gov