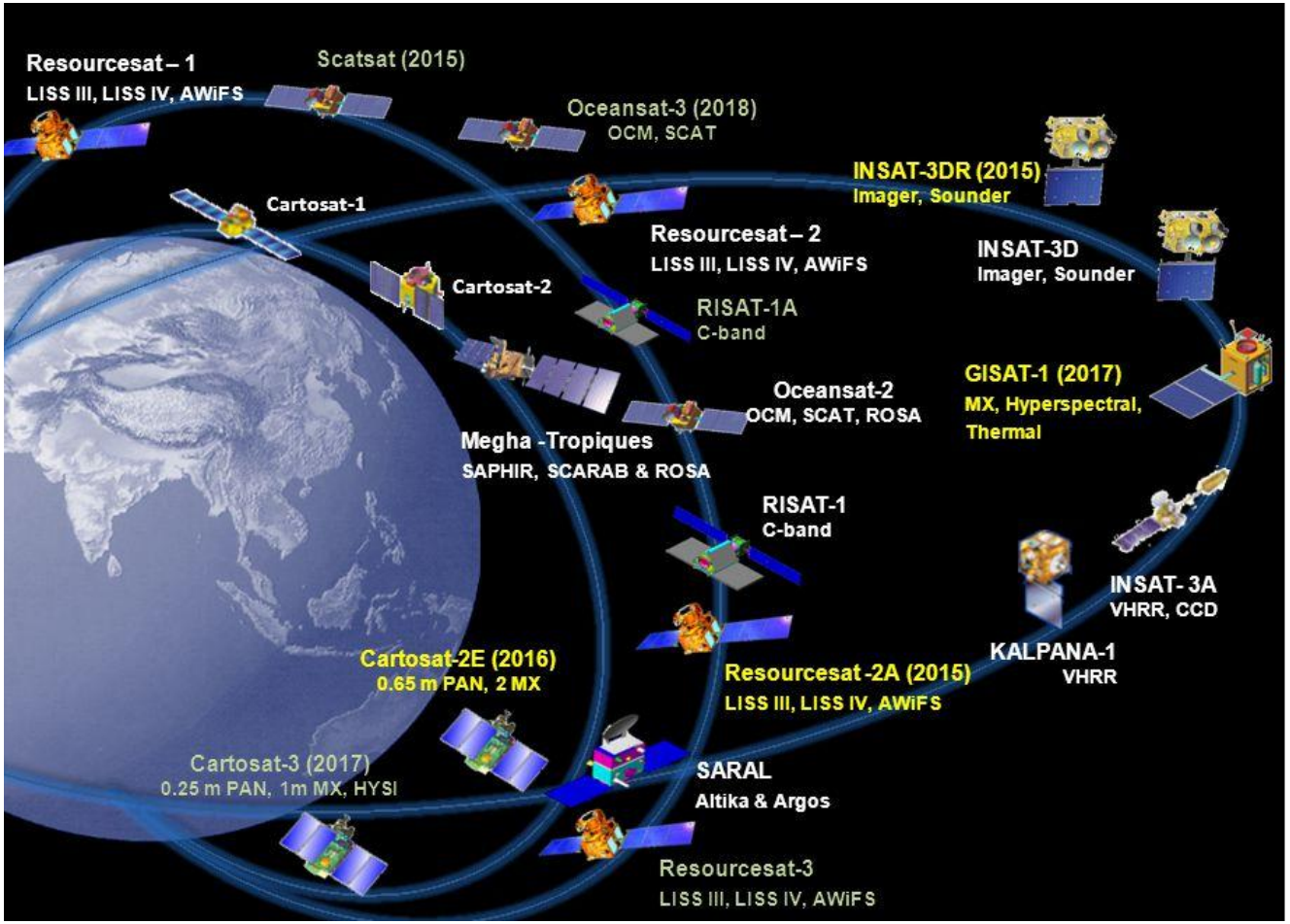
Introduction to Google Earth Engine



- Large number of data sources e.g. ISRO alone specifies 130 missions (as of 2022)
- 10-20 TB /day
- Requirement for high accuracy of models.
- Requirement for real time analysis.

Source: ISRO website, 2015

How can we work with the data

- High end PCs/Workstations with multiple cores.
- Graphics Processing Units
- HPC, Parallel, clustered or distributed systems

- High costs for installation and maintenance.
- Not suitable for small organizations or individuals.

CLOUD COMPUTING: Scalable, no hassle of downloading data, parallelized set up, pocket-friendly.
Eg: AWS, Microsoft Planetary Computer, Digital Earth, Google Earth Engine

Contents lists available at ScienceDirect

Remote Sensing of Environment

journal homepage: www.elsevier.com/locate/rse

Google Earth Engine: Planetary-scale geospatial analysis for everyone

Noel Gorelick [a,*], Matt Hancher [b], Mike Dixon [b], Simon Ilyushchenko [b], David Thau [b], Rebecca Moore [b]

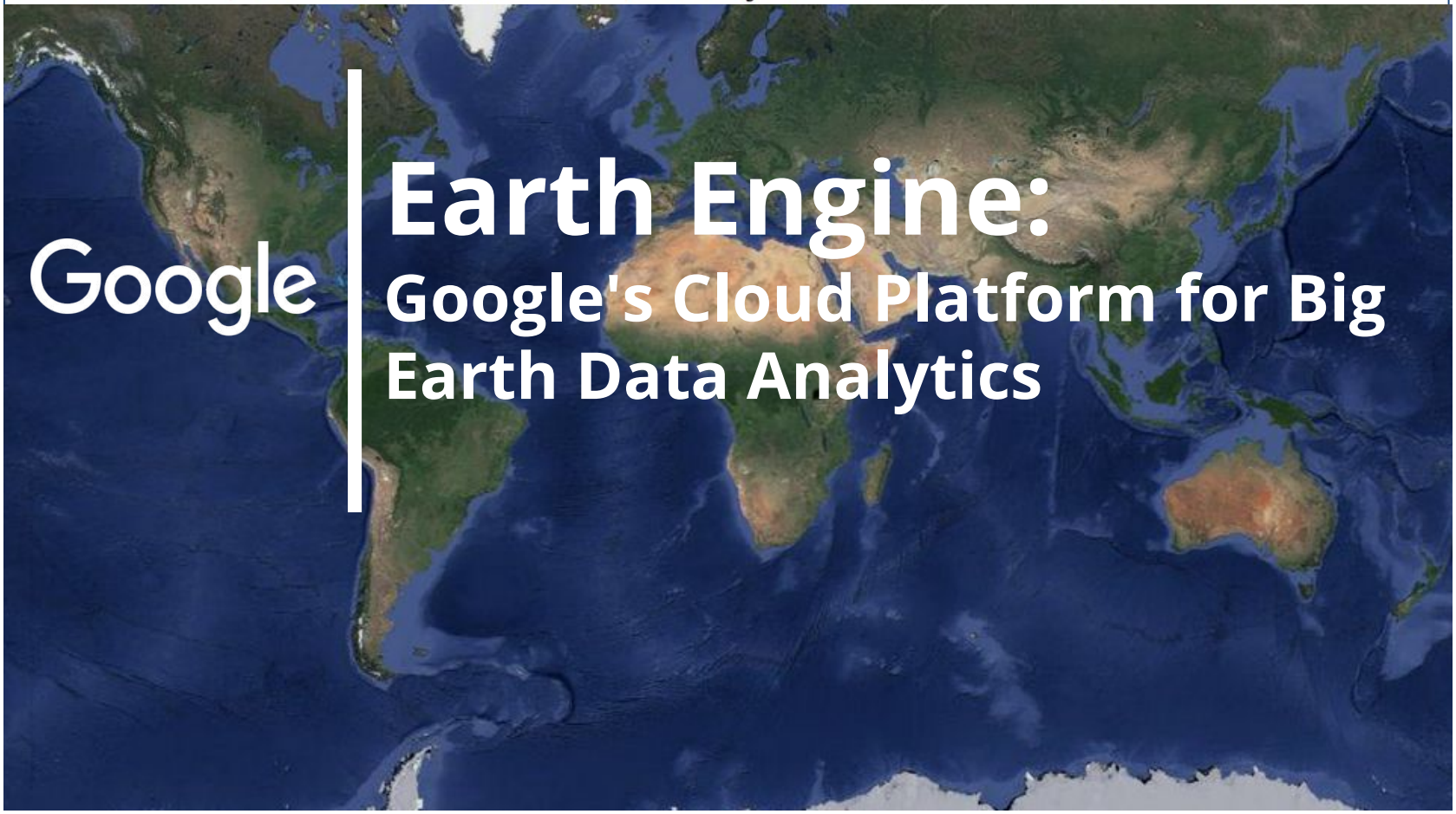


Image courtesy: Gennadii Donchyts

The Earth Engine Public Data Catalog

Landsat 4, 5, 7, 8
Raw, TOA, SR, …

MODIS
Daily, NBAR, LST, …

Terrain
SRTM, GTOPO, NED, …

Land Cover
GlobCover, NLCD, …

Atmospheric
NOAA NCEP, OMI, …

… and many more, updating daily!

> 200 public datasets

> 20 million images

> 35 years of data

> 4000 new images every day

> 20 petabytes of data

> 12 quadrillion pixels

"Often it turns out to be more efficient to move the questions to the data than to move the data to the questions." -Jim Gray in the Fourth Paradigm

Stats and Image courtesy:
Gennadii Donchyts, Nick Clinton

Features of Earth Engine

Data Types

- **Rasters:** Image, Image Collection
- **Vectors:**
  - Geometry: Point, Line, Poly
  - Feature: Geometry and attribute information
  - Feature Collection: set of features
- **Projection:** CRS, transforms. Default is pseudo Mercator EPSG 3857.
- **Model:** Trained ML models
- **Common programming datatypes:** Integer, String, Date, Array, Dictionary etc

Algorithms

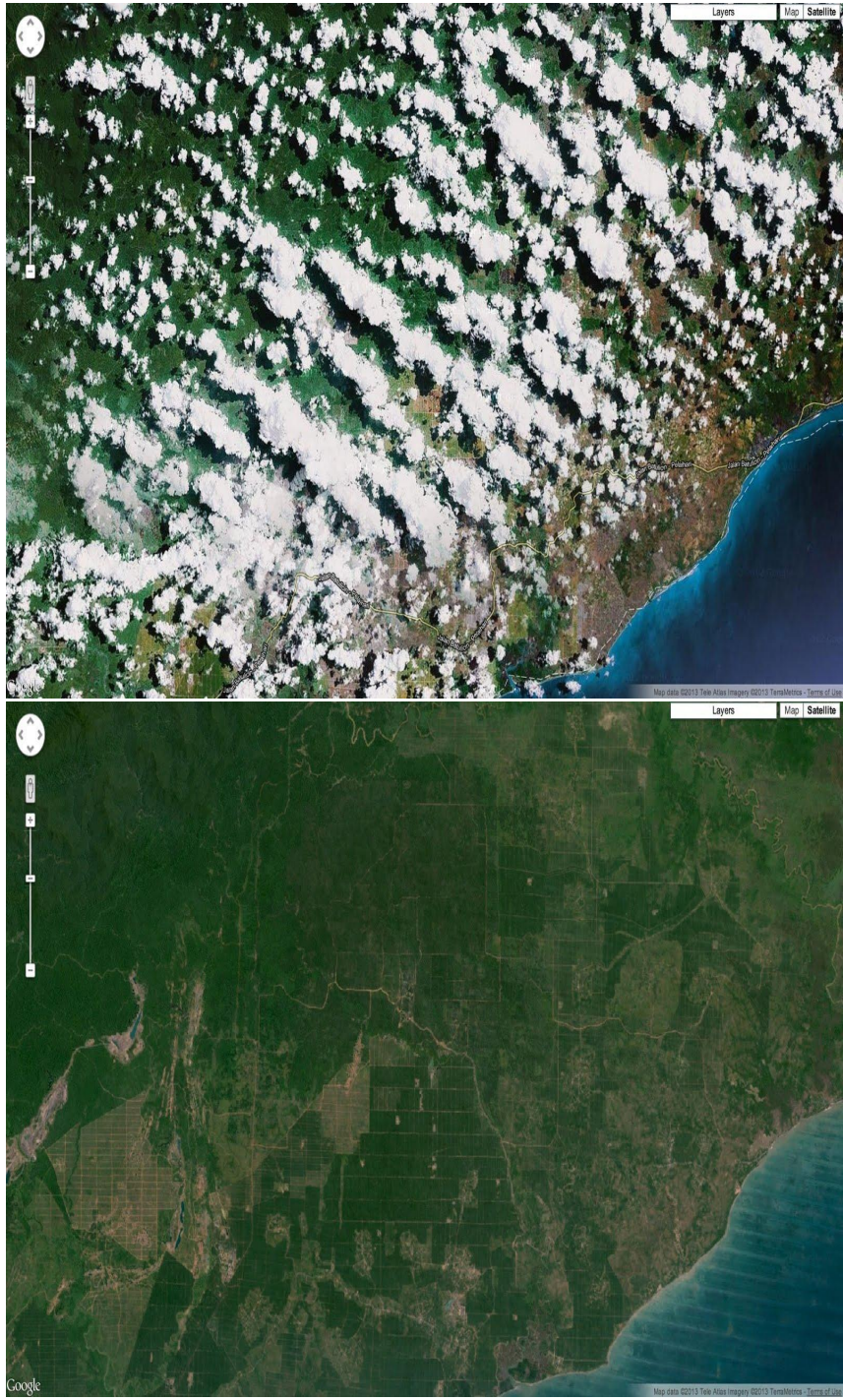
- **Data preprocessing:** Filters, radiance calibration, f-mask filtering, hillshade, slope, aspect
- **Feature Extraction:** Sobel, Roberts, Prewitt, Laplacian
- **ML tools:** CART, Random Forest, Gradient Boosted, SVM, SNIC, K-Means, Accuracy Assessment tools. More from sklearn and TensorFlow.
- **Reducers:** Image collection to image, feature collection to feature/image, Image to number using aggregation such as mean, median, minMax, first, etc.
- **Visualization:** Maps, Charts, UI

Capabilities of Earth Engine

- Optimized data download via Google Drive/Cloud Storage
- Large-scale time lapse visualization
- Earth Engine Explorer
- JavaScript API emphasis using Code Editor
- Python API to integrate with Machine Learning Workflows
- QGIS integration for mapping and data representation
- Google Earth Engine Apps

Optimized data download and Earth Engine Explorer

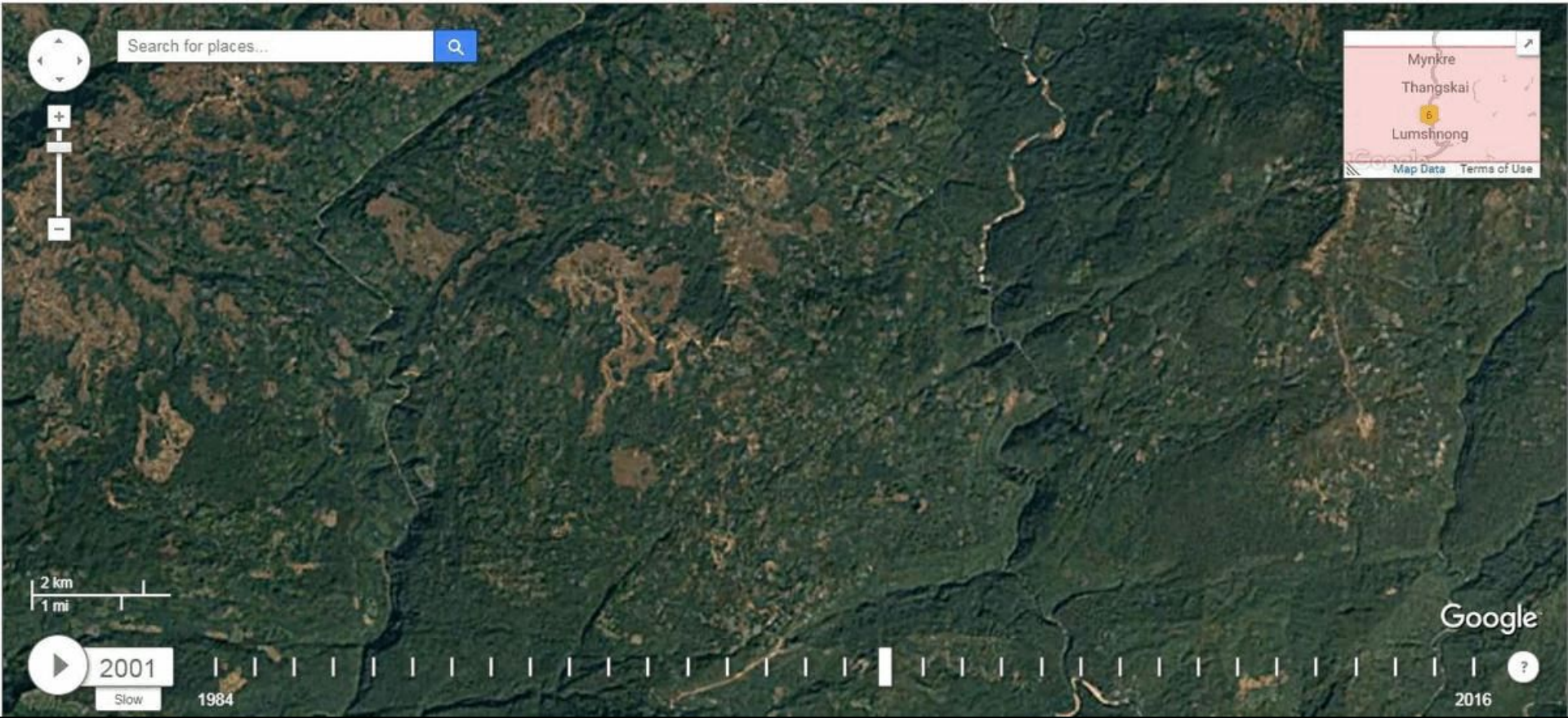
- Access and visualize datasets across locations and time steps desired at varying levels of processing
- Access large number of datasets across the domains
  - Climate and weather
  - Satellite imagery
  - Terrain
  - Land Cover
  - Other geophysical data
- Cloud-free data generation through temporal merging
- Code-free processing tools including classification



Large-scale time lapse visualization

Right: GIF of monthly mean NDVI visualization for NER from 2018 to 2019

Bottom: Quarry land time lapse at Lumshnong, Meghalaya



JavaScript API

All your saved and example scripts

Documentation

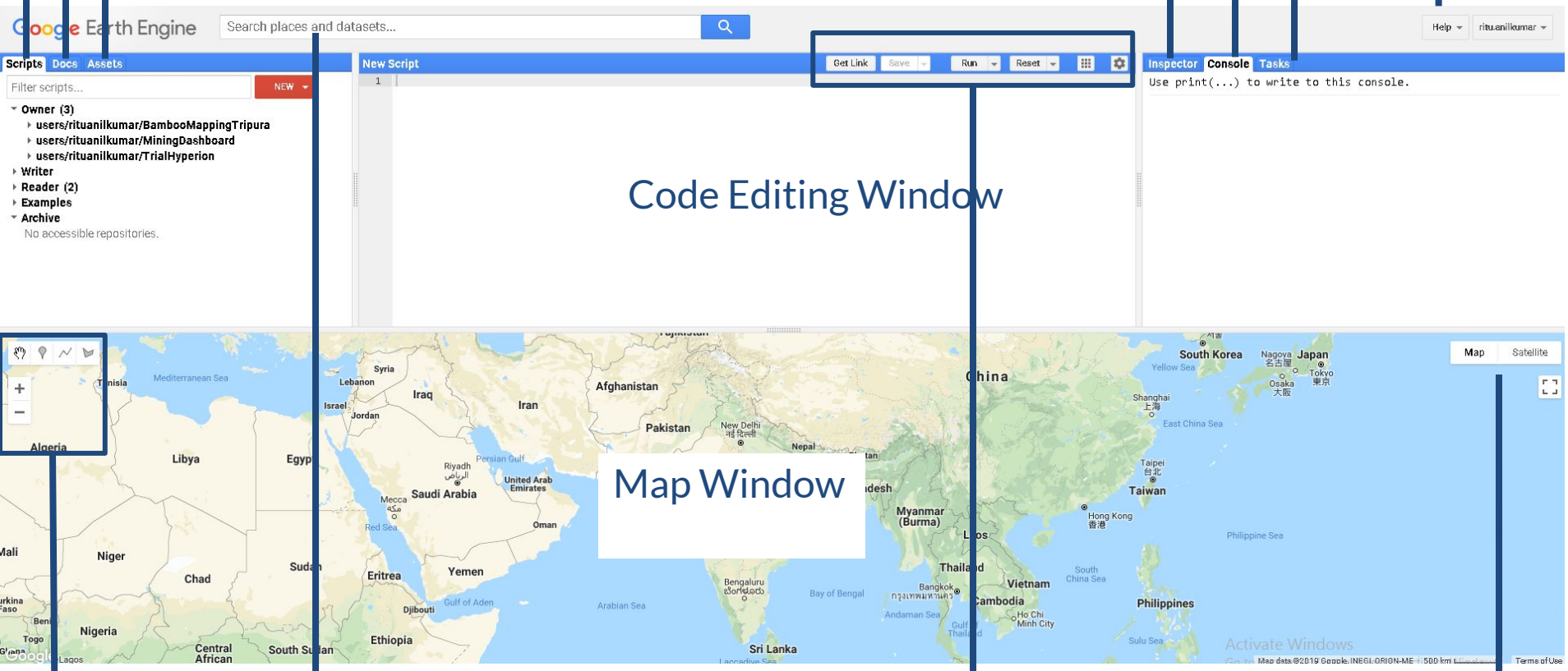
All your uploaded assets

Task status: E.g. Exports

Console output

Inspecting map elements

User Guide/ Discussions Forum



Code Editing Window

Map Window

Tools for creation of vectors, pan, zoom

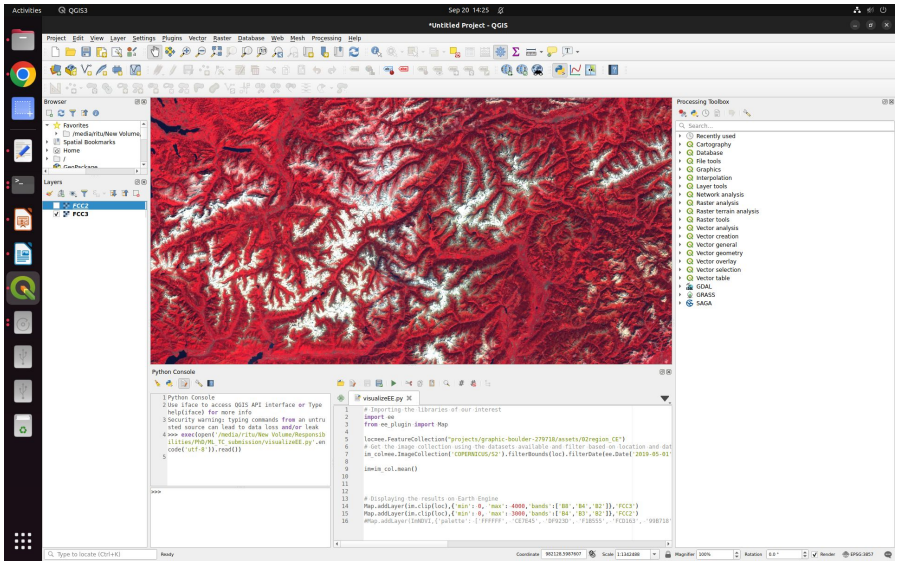
Search box for all datasets

Link sharing, saving, running etc

Map/Satellite View Toggle

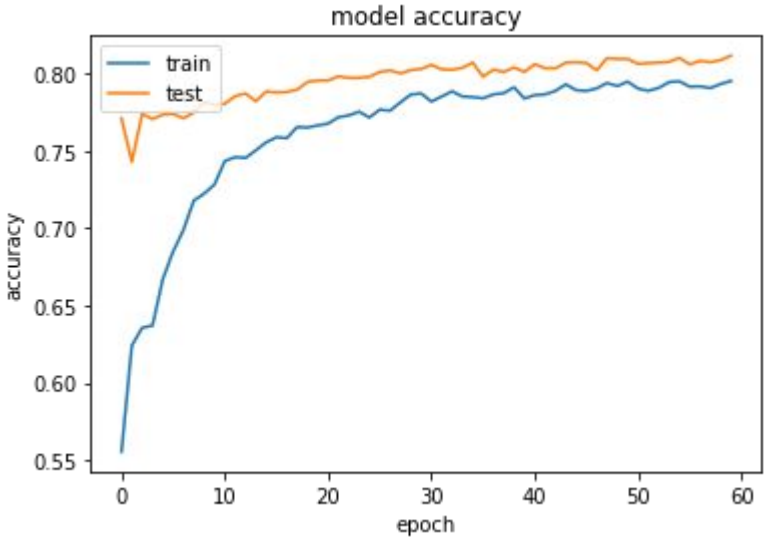
Python API and Integration with GIS Tools

- Visualize and create maps on QGIS using Python interface and Earth Engine plugin
- Example project: forest fire burnt patch estimation with automated buffering and quantile derived thresholding.

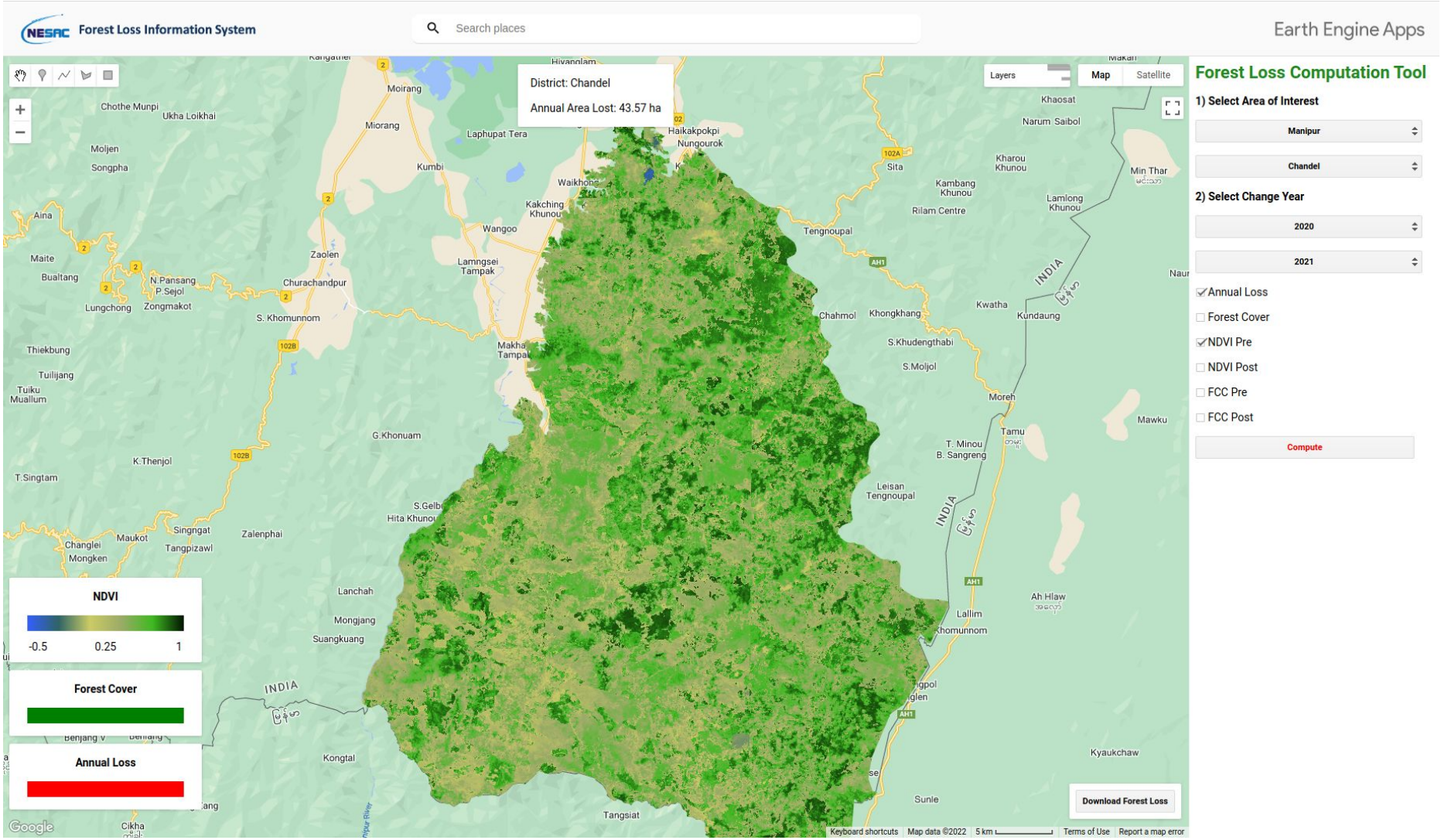


Python API and Integration with ML Libraries

- Integration with other Python statistical and machine learning libraries
- Tested with TensorFlow for the DTDI project extended with Sentinel 2 and PROBA V datasets



Web Apps using App+Earth Engine



Forest Loss: https://**nilaynishant**.users.earthengine.app/view/forest-loss

Projects Implemented

1. Open Cast Mine Identification in Meghalaya
2. Vegetation loss associated with Coal and Limestone Mining, Jaintia Hills
3. Monitoring of Strip Mining activity in Saleki PRF, Assam
4. Burnt area assessment for Manipur
5. Pan-NE Forest Loss Studies
6. Glacier change studies, Sikkim
7. Identification of permanent river islands in Assam
8. LULC mapping for parts of Assam
9. Bamboo dominated area mapping for Tripura
10. Flood mapping using Sentinel 1 (by Nilay Nishant)

Techniques Implemented

- Cloud free composite generation
- Landsat 7 destriping
- Classification (incl ML pipelines using Python)
- PCA based dimensionality reduction
- Textural features using GLCM and OBIA image segmentation
- Charting and time series visualization

JavaScript Hands on: Click Here

Python Hands-on: Click Here

Some Resources: Click Here

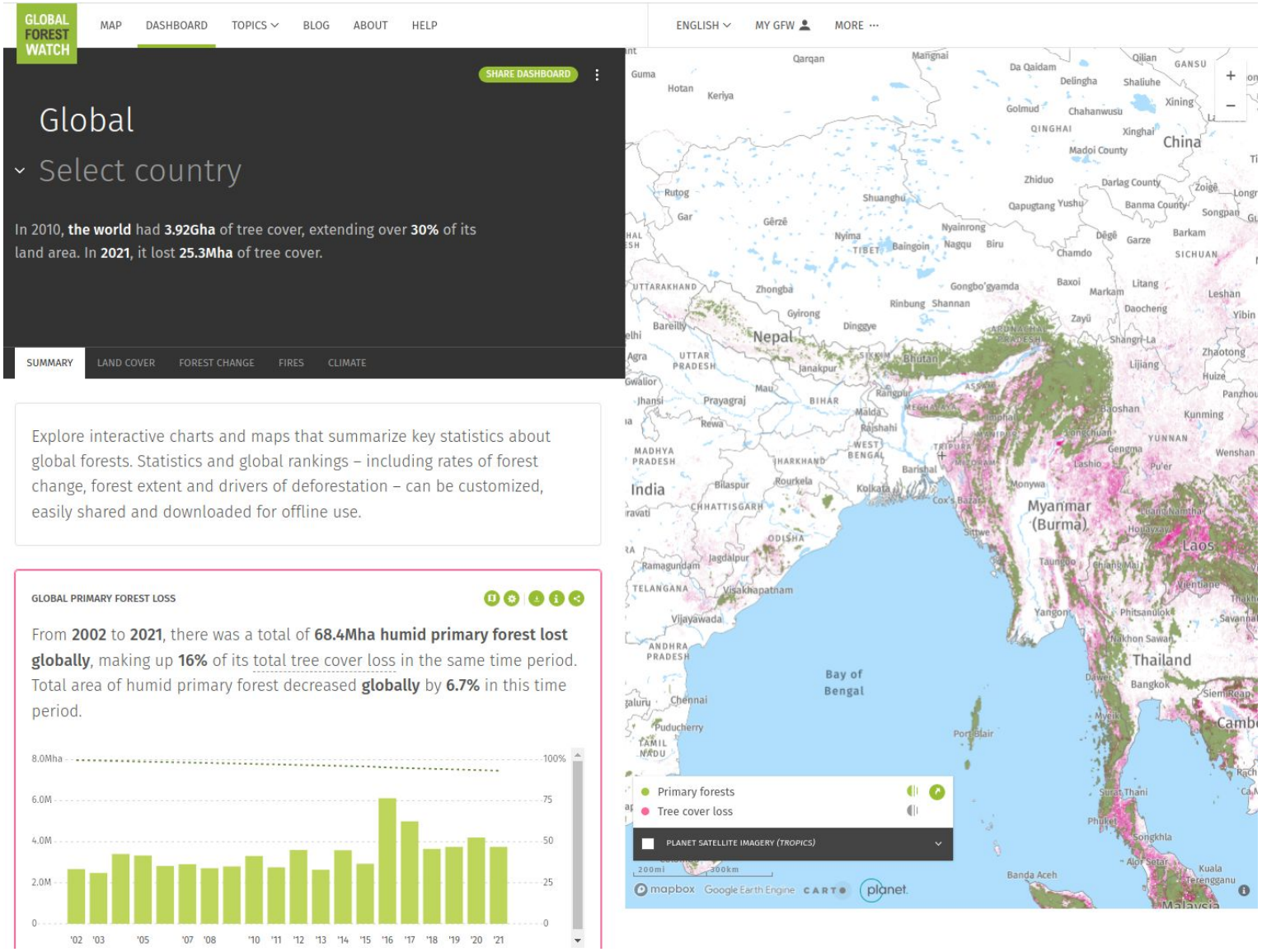
Here

Here

and here

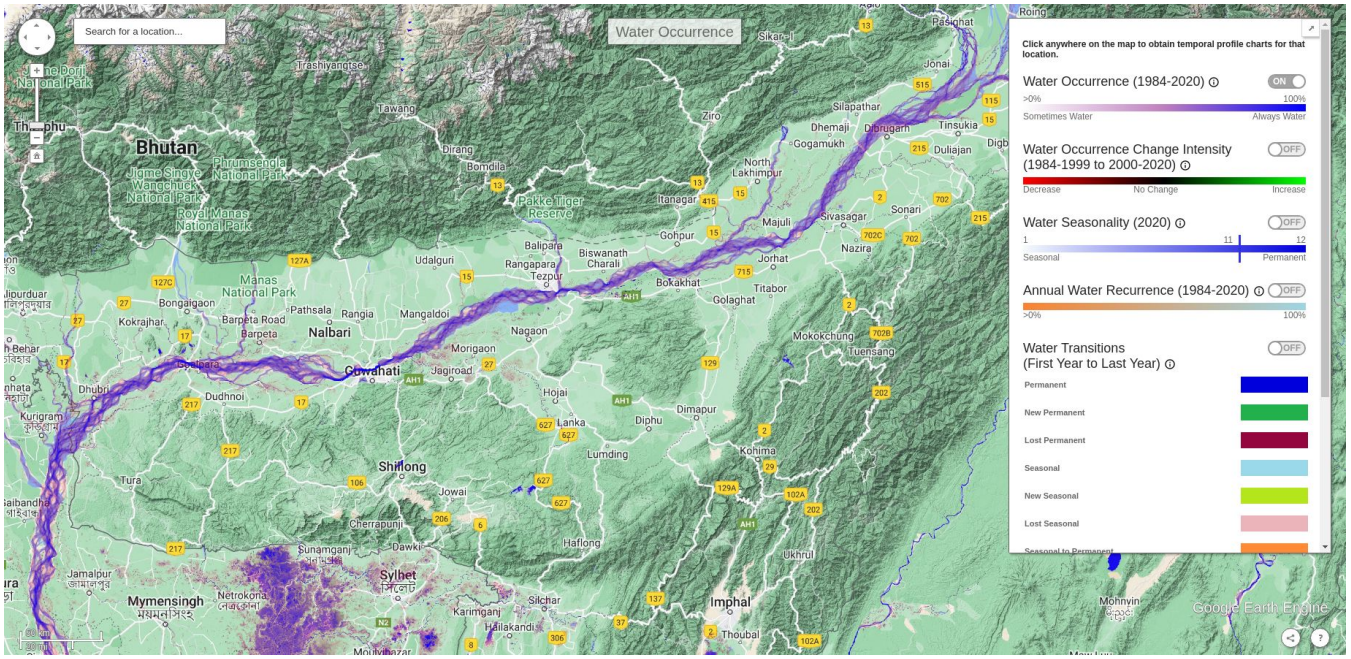
SOME COMMON STUDIES USING EARTH ENGINE

Global Forest Watch



- Forest loss statistics and visualization globally or by country
- Webpage: https://www.globalforestwatch.org/

Global Surface Water



- View water bodies and understand their seasonal behaviour
- Webpage: https://global-surface-water.appspot.com/map

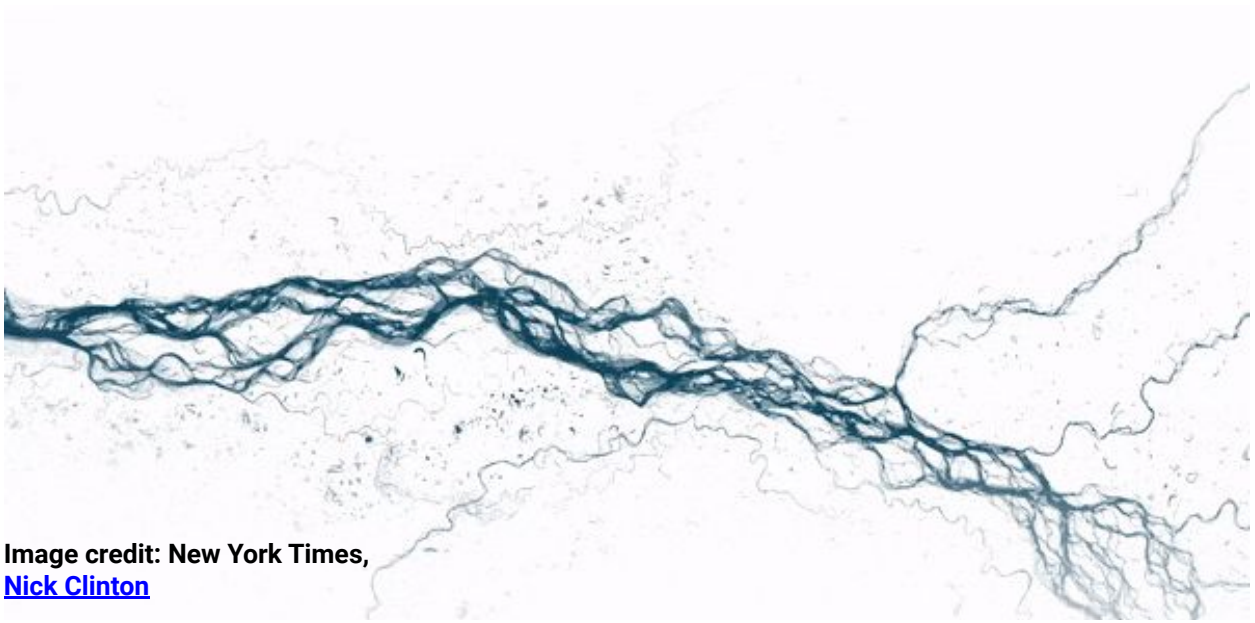
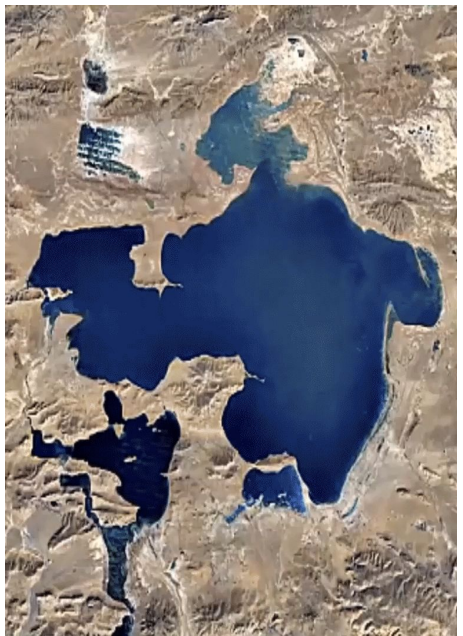
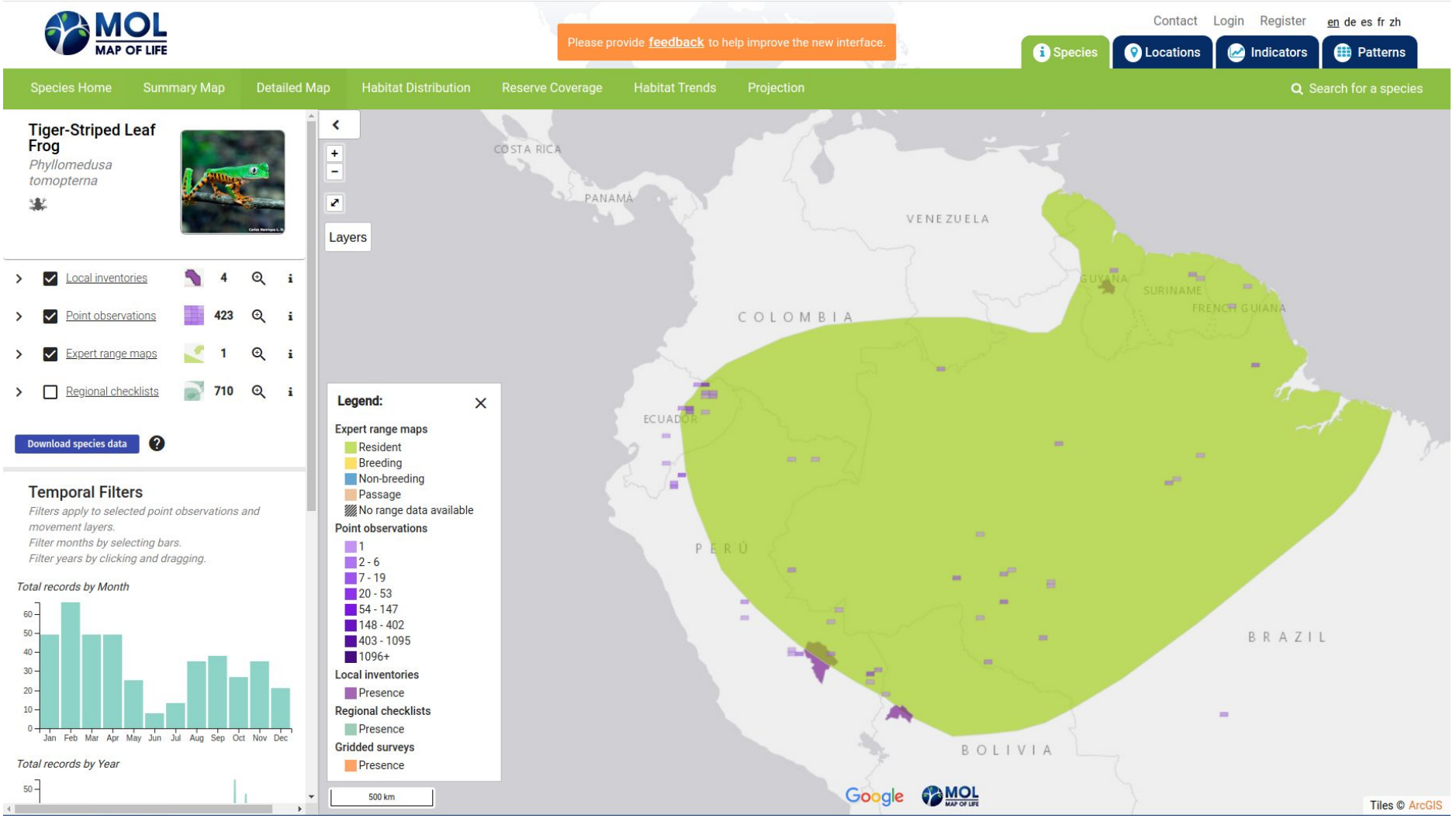


Image credit: New York Times, Nick Clinton

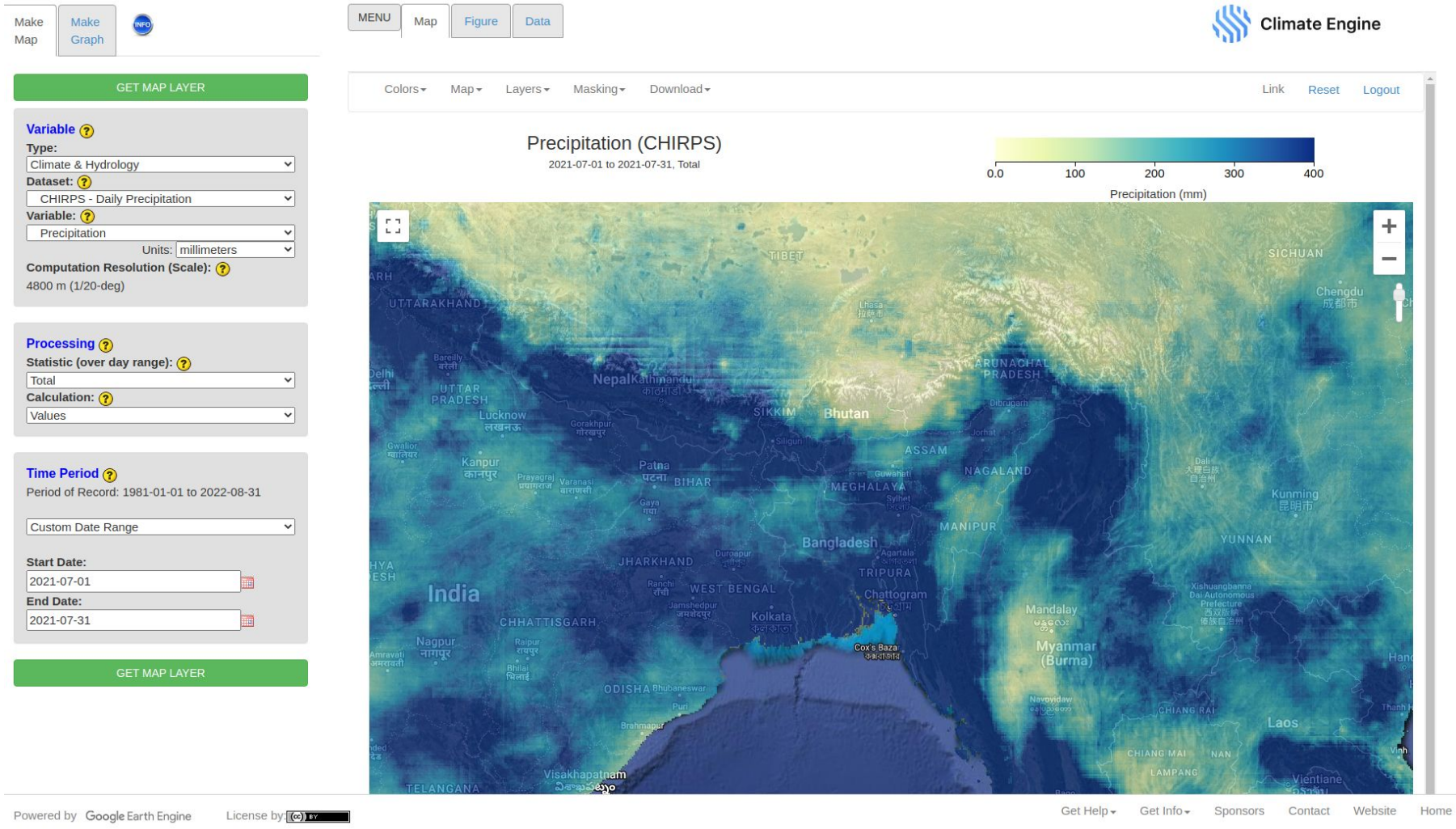


Map of Life



- View details of species distribution, status and sightings
- Webpage: https://mol.org/species/

Climate Engine



- View weather datasets and perform basic time series analysis
- Webpage: https://app.climateengine.com/climateEngine

THANK YOU