LINDEX, an End-to-End Landsat-8 Timeseries Index Processing Framework

Travis Simmons[1] and James Deemy[1]

[1]Department of Natural Sciences, College of Coastal Georgia, Brunswick GA, USA 31525

November 22, 2022

Abstract

As Earth's ecological landscape continues to change, it will become increasingly important to understand how it has changed, and how it may change in the future. Freely available multispectral remote sensing datasets, such as the Landsat-8 dataset accessed through the USGS EarthExplorer tool, provide large scale, high resolution satellite imagery that can be leveraged by researchers across scientific disciplines for timeseries index analysis. LINDEX provides an extensible framework that automates Landsat-8 timeseries index analysis, resulting in an average ninety-four percent reduction in overall processing time compared to hands-on methods. Traditionally, in order to make use of this historical data, researchers must acquire the data, uncompress each scene, crop each scene to their region of interest, sort each cropped scene by date and cloud cover, and then either use GIS tools to run index analyses or code the index analyses themselves. This process is time consuming, requires extensive computational knowledge, and is prone to human error. In order to address these challenges, we developed LINDEX, a fully containerized end-to-end extensible processing framework for Landsat-8 timeseries index analysis. LINDEX is open source and leverages open source python packages to automate decompression, cropping, cloud cover detection, sorting by date, and index analysis for Landsat-8 data while also providing a customisable and growing library of eleven ecologically useful indices including NDWI, NDMI, NDSI and NDGI. LINDEX is designed to work synergistically with the EarthExplorer Bulk Download Application as well as QGIS, providing a bridge from download through analysis. The LINDEX framework uses a well defined methodology for incorporating custom indices into your workflow, making LINDEX a useful tool for researchers interested in exploring any Landsat-8 multispectral index while saving time, and reducing error.

LINDEX, an End-to-End Landsat-8 Timeseries Index Processing Framework

Travis T. Simmons, James B. Deemy

Department of Natural Sciences, College of Coastal Georgia, Brunswick GA, USA 31525

Background

- Timeseries index analysis of satellite data can be used to track changes in a variety of physical and ecological parameters over time.
- Major limitations in the use of bulk datasets and satellite image archives are the time intensive data decompression, extent processing, cloud detection, and index analysis.
- LINDEX was developed as a single tool to address these limitation by taking advantage of open-source tools and packages to automate data preparation and analysis. This results in a 94% reduction in analysis time and the ability to tailor the index used to meet unique research goals.

Objectives

1. Create a tool to assist researchers in doing timeseries index analysis by automating the decompression, cropping, and analysis of Landsat data.
2. Create a customizable framework for researchers to deploy any index of interest.
3. Containerize the tool for easy deployment in any local or cloud computing environment and to increase analysis reproducibility.

Table 1: Technologies and Python packages used.

Technologies	Python Packages
Docker	OpenCV
Singularity	GDAL
QGIS	Rasterio
Lat Lon Tools	MoviePy
USGS Earth Explorer	Matplotlib
USGS Bulk Download Application	Xtarfile

Open-Source Code:

https://github.com/Travis-Simmons/LINDEX

Container hosted at:

https://hub.docker.com/repository/docker/travissimmons/lindex

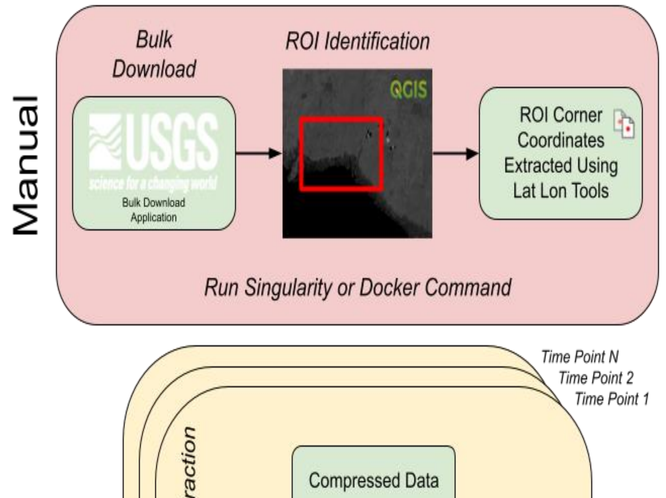

Figure 1: LINDEX deployment workflow.

Table 2: Available indices.

Index	Formula
Normalized Difference Water Index (NDWI)	(NIR-SWIR)/(NIR+SIWR)
Normalized Differencce Vegetation Index (NDVI)	(NIR-RED)/(NIR+RED)
Enhanced Vegetation Index (EVI)	G*((NIR-RED)/(NIR+C1*R-C2*BLUE+L))
Advanced Vegetation Index (AVI)	(NIR*(1-RED)*(NIR-RED))^(1/3)
Soil Adjusted Vegetation Index (SAVI)	((NIR-RED)/(NIR+RED+L))*(1+L)
Normalized Difference Moisture Index (NDMI)	(NIR-SWIR)/(NIR+SWIR)
Moisture Stress Index (MSI)	MidIR/NIR
Green Chlorophyll Index (GCI)	(NIR)/(GREEN)-1
Normalized Burn Ratio (NBR)	(NIR-SWIR)/(NIR+SWIR)
Bare Soil Index (BSI)	((RED+SWIR)-(NIR+BLUE))/((RED+SWIR)+(NIR+BLUE))
Normalized Difference Snow Index (NDSI)	(GREEN-SWIR)/(GREEN+SWIR)
Normalized Difference Glacier Index (NDGI)	(NIR-GREEN)/(NIR+GREEN)
Custom	Custom

```python
def custom_index_template(date_folder):

    # replace the underscores with the band you need, repeat as necissary
    band_ = glob.glob(os.path.join(date_folder, '*B_.TIF'))
    b_ = rio.open(band_[0])

    # After adding in each band you will be using,
    # rename then as their common name eg: nir, red, green ...
    band_name = b_.read()
    band_name = band_name.astype(float)

    # Replace index name with your index
    # do the raster math with the common names
    index_name = (band_name-band_name)/(band_name+band_name)

    # Close all the bands, repeat as necissary
    b_.close()

    # Scroll down and find 'index dict'
    # to add your index to the options before running
    return index_name
```

Figure 2: Custom index template.

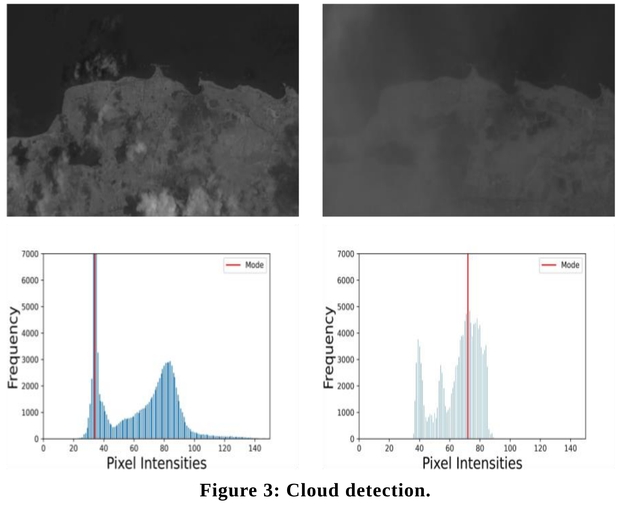

Figure 3: Cloud detection.

Use Cases

- This tool can be used to conduct any index analysis of any region at any time period covered by the publicly available Landsat-8 dataset.
- NDWI can be used to detection of seasonal wetlands and episodic flow.
- NDVI can be used for island migration tracking, crop inventory, and forest canopy tracking.
- NBR can be used for wildfire tracking.
- NDSI and NDGI can be used to track snowfall, sea ice, and glaciers.

Current Limitations

- Refinement of cloud detection approach is needed to increase the accuracy of image sorting.
- LINDEX only has one dependency, Docker or Singularity.

Future Direction

- Create a stand-alone GUI to remove dependencies.
- Extending LINDEX to other satellite datasets.
- Automatic task distribution using Makeflow and Work Queue.
- USGS Machine-to-Machine API integration.

Acknowledgements

We would like to thank the PhytoOracle team at The University of Arizona for supplying the template for our dockerfile (https://github.com/phytooracle). Many thanks go to the USGS for the provision of public data. We also thank the Coastal Georgia Department of Natural Sciences and the Coastal Georgia Service-Learning Office for supporting our participation in this conference. Additionally, presentation of this poster would not be possible without support from the AGU student travel grant. Finally, we would like to thank the conference hosts for providing the space and the conveners for organizing this session.