Geoprocessing in python (SS 2019)

Conservation Biogeography Lab & Earth Observation Lab,
Geography Department, Humboldt-University Berlin
Matthias Baumann, Dirk Pflugmacher

Assignment IV

Due date	Sunday, 19.5.2019 10pm
Submission form	1. Questions in relation to the scripting task within a moodle-quiz. 2. py-Script
Evaluation criteria	• Correctness of the asked results (50%). • Functionality of the py-script (50%).

Goal of this assignment

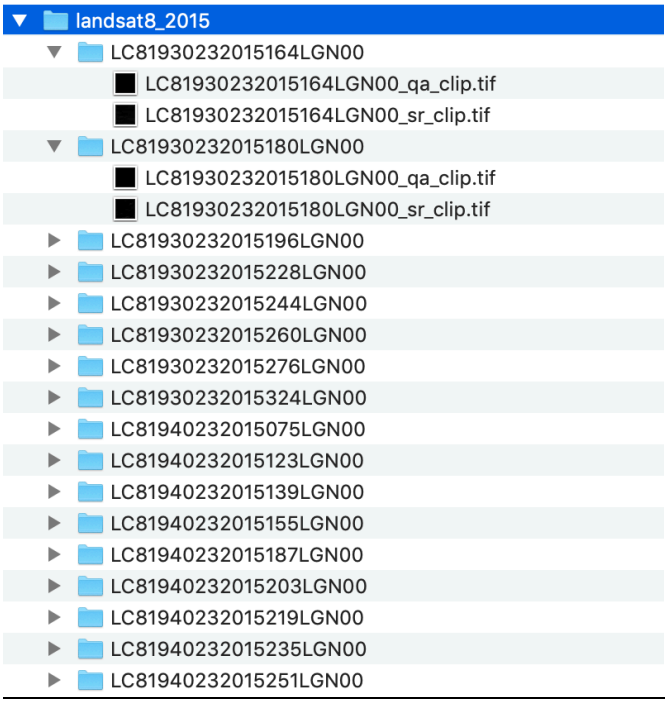
- The Basics of NumPy Arrays
- Computation on NumPy Arrays: Universal Functions
- NumPy Aggregations: Min, Max, and Everything In Between
- NumPy Computation on Arrays: Broadcasting
- NumPy Comparisons, Masks, and Boolean Logic
- GDAL write multi-band raster

Exercise I

Optical satellite images can contain clouds and cloud-shadows that need to be masked out prior to analysis. In many regions of the world, frequent cloud cover makes cloud-free images rare throughout the year. One way to alleviate this issue is to aggregate time series of multiple images by calculating the mean pixel values by band over a specified time period. The goal of this exercise is to calculate a mean Landsat-8 image using data from the year 2015.

You are supplied with overlapping spatial subsets (1000 x 1000 pixels) of 17 Landsat-8 images. Each image is stored in a separate folder, and it consists of two GeoTIFF files:

> 1) SR raster - *sr_clip.tif: six band/layer raster of surface reflectance in the blue, green, red, near-infrared, shortwave-infrared-1, and shortwave-infrared-2 and
>
> 2) QA raster - *qa_clip.tif: one single band/layer raster with the following pixel codes: 0 => clear land pixel, 1 => clear water pixel, 2 => cloud shadow, 3 => snow, 4 => cloud, 255 => fill value.



1. Build mask: Create a function that reads a single QA file and returns a Boolean numpy array (True: good observations, False: invalid observations). Good observations are clear land, water, and snow pixels (See the codes in the exercise description).

2. Mask image: Create a function that takes the filename of a single SR image and outputs a numpy array containing the masked-out image data. Inside the function, you should identify the corresponding QA file. Use the function from task 1 to retrieve the correct mask array.

3. Mean image: Create a function that takes SR file names as input and outputs a mean SR image, i.e., a six-band image where each band and pixel represent the mean across time. Utilize the functions you created in task (1) and (2). Hint: A NaN is a special value for float arrays only.

4. Max NDVI Composite: Create a function that takes SR file names as input and write a maximum NDVI composite as GeoTIFF. In a maximum NDVI composite each pixel contains the band values that correspond with the date of maximum NDVI. Hence, each pixel will represent the original band values that were recorded at the date of maximum vegetation greenness.