**Data Readme**

Last updated: FC 7/24/2020

This document outlines the data workflow steps

* Data to download is selected from the NASA LAADS website:  
  <https://ladsweb.modaps.eosdis.nasa.gov/search/>  
  We are using the MOD09.GA (Terra) and MYD09.GA (Aqua) datasets. Right now, we are using the tile H16 V7. Both Terra and Aqua datasets are downloaded and each subsequent code works on both datasets.
* If you’ve created an account (free), you will receive an email with a “wget” command to download the data. This is the easiest way to download the data. When downloaded in this way, it is in the HDF format.
* The code P0\_process\_modis\_data.py is used to extract the bands from the hdf data format and output them into 3-channel and 7-channel numpy arrays. For this step, bad pixels in the output numpy arrays are given the value “-28672” and are output into a separate file. Additionally, three images are created by this code: [1] A true-color image using bands 1, 4, and 3, [2] an image of the individual channels with a default colorbar, and [3] an image of the individual channels with a colorbar maxed out at a value of 3e3 to bring out dust storm features. Note, this vmax only applies to plotting and not the numpy arrays. This code is parallelized using the joblib library and can use a number of CPU cores enabled by the user.
* The code P1\_resample\_normalize takes the 3-channel output produced by P0 and downscales the images to a length/width in pixels provided by the user in a vector at the beginning of the “main” code. For right now, the length/widths tested are 224, 448, and 672 pixels. Data is loaded, the “-28672” values are converted to np.nan, and the cv2 library is used to downscale the images to the desired size. In addition, the cv2 library also interpolates across the np.nan values. Once resized, each band within the image is saved into a new numpy file, and both individual band plots and a full-color image are also saved. This code is parallelized using the joblib library and can use a number of CPU cores enabled by the user.
* Code P2 takes the data produced by P1 and computes the pixel-wise mean, median, and standard deviation
* Code P3 removes produces two products: [1] median removed data and [2] mean-removed data divided by the standard deviation