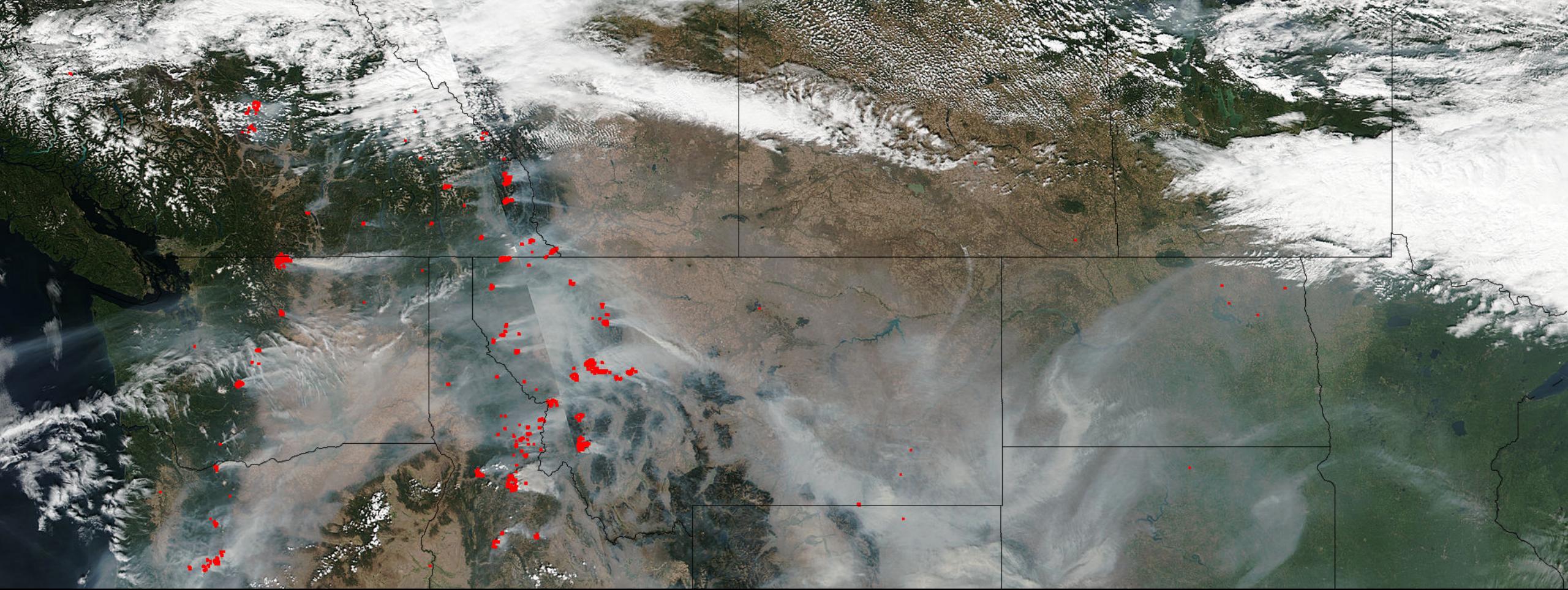


Read, Map, and Extract MODIS Aerosol Data Using Python Scripts

Melanie Follette-Cook and Pawan Gupta

Satellite Remote Sensing of Dust, Fires, Smoke, and Air Quality, July 10-12, 2018

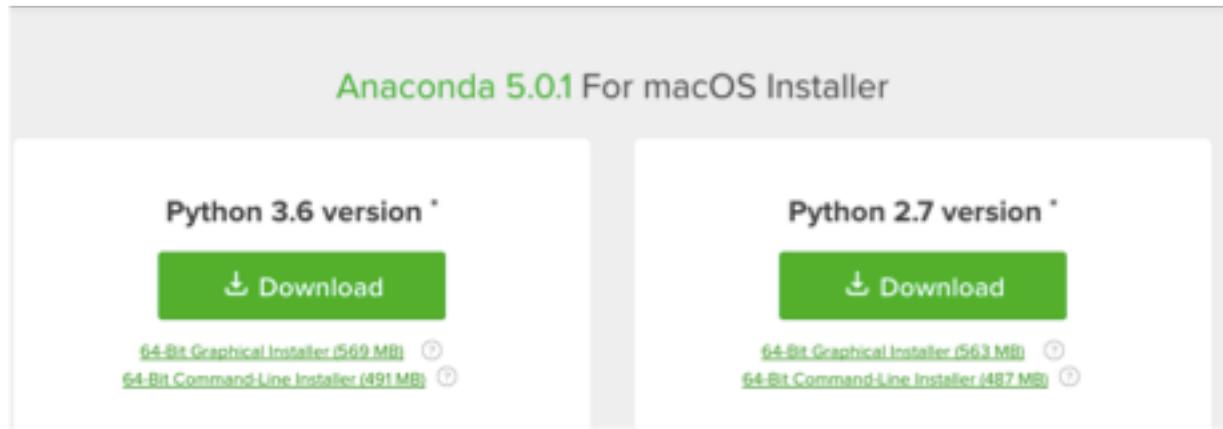




Computer and Python Requirements

Computer Requirements

- Install Python 2.7 using Anaconda
- Install all required python packages
 - Package List (right)
- Test Python and package installations using the following Python test code
 - [test_python.py](#)
- Download MODIS Data and Python Codes using the following link
 - [ARSET LINK ZIP FILE](#)
- For more detail on the code, visit:
<https://arset.asfc.nasa.gov/airquality/python-scripts-aerosol-data-sets-merra-modis-and-omi>



- Python package list:
 - pyhdf
 - numpy
 - sys
 - mpl_toolkits.
basemap
 - matplotlib
 - linearSegmented
Colormap
 - h5py
 - time
 - calendar



Python Test

- Open the spyder editor inside Anaconda
- Open **test_python.py**
- Make sure the directory has the Python code and HDF file
- Open the **ipython** console in the spyder
- Run the code using the **green arrow** on the top
- Output should be an image as shown

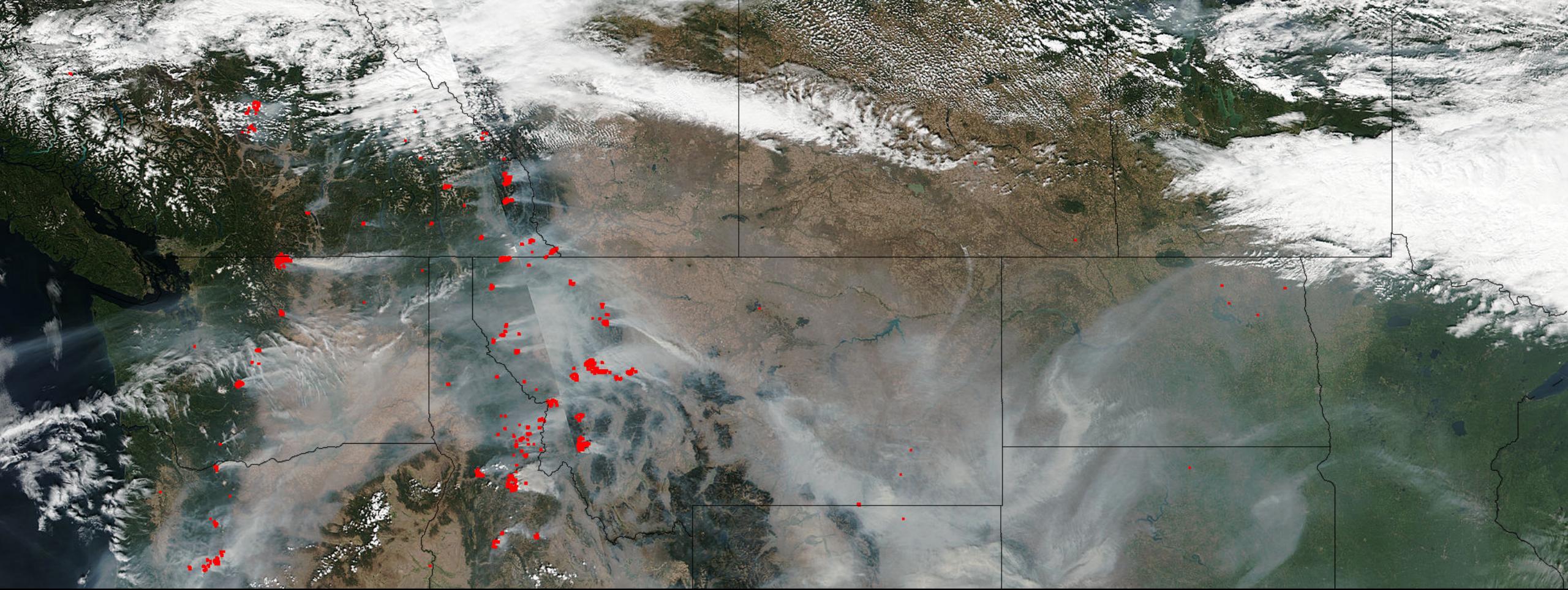
The screenshot shows the Spyder IDE interface. On the left, the 'Editor' tab displays the Python script `test_python.py`. A green arrow points to the run button at the top of the editor window. A blue arrow points to the line of code `FILE_NAME='MYD04_L2.A2017249.2105.006.2017250160535.hdf'`, which is highlighted in blue. On the right, the 'Console' tab shows the command-line interface with the code being run and its output. The output includes a warning about the `ishold` function being deprecated and a map titled 'MYD04_L2.A2017249.2105.006.2017250160535 Image Optical Depth_Land_And_Ocean'. The map shows optical depth data over a geographic area from 35°W to 110°W and 35°N to 50°N. A color bar on the right indicates depth values from -6 to 5. A blue arrow points to this map, labeled 'HDF file'. Another blue arrow points to the map itself, labeled 'output'.

```
#!/usr/bin/python
...
Module: read_and_map_mod_aerosol.py
-----
Disclaimer: The code is for demonstration purposes only. Users are responsible to check for accuracy.
Author: Justin Roberts-Pierrel, 2015
Organization: NASA ARSET
Purpose: To extract AOD data from a MODIS HDF4 file (or series of files) and create a map of the same.
See the README associated with this module for more information.
...
import necessary modules
from pyhdf import SD
import numpy as np
from mpl_toolkits.basemap import Basemap, cm
import matplotlib.pyplot as plt
import sys
import h5py
import time
import calendar
...
FILE_NAME='MYD04_L2.A2017249.2105.006.2017250160535.hdf'

hdf=SD(FILE_NAME)
x,y = hdf.get_lat_and_lon()
lat = hdf.select('Latitude')
latitude = lat[:]
min_lat=latitude.min()
max_lat=latitude.max()
lon = hdf.select('Longitude')
longitude = lon[:]
min_longitude=longitude.min()
max_longitude=longitude.max()
SDS_NAME='Image_Optical_Depth_Land_And_Ocean'
sdshdf.select(SDS_NAME)
#open scale factor for AOD SDS
attributes=sds.attributes()
scale_factor=attributes['scale_factor']
#get valid range for AOD SDS
rangen SDS.getrange()
min_rangen=min(rangen)
max_rangen=max(rangen)
#open SDS data
datasds=SDS()
#get data within valid range
valid_data=datasds.read()
valid_data[valid_data < min_rangen] = np.nan
valid_data[valid_data > max_rangen] = np.nan
...
In [1]: runfile('/Users/gupta/Desktop/CA_TRM/test_python.py', wdir='/Users/gupta/Desktop/CA_TRM')
/Users/gupta/.python/anaconda/lib/python2.7/site-packages/mpl_toolkits/basemap/__init__.py:3413: MatplotlibDeprecationWarning: The ishold function was deprecated in version 2.0.
b = ax.ishold()
/Users/gupta/.python/anaconda/lib/python2.7/site-packages/mpl_toolkits/basemap/__init__.py:3422: MatplotlibDeprecationWarning: axes.hold is deprecated.
See the API Changes document (http://matplotlib.org/api/api\_changes.html)
for more details.
ax.hold(b)

MYD04_L2.A2017249.2105.006.2017250160535
Image Optical Depth_Land_And_Ocean
In [2]:
```





Know Your Data

Understanding a MODIS File Name

Level 2, 10 km, Aerosol Product

Product Name:

- Terra: MOD04
- Aqua: MYD04



HDFLook, Panoply, IDL, Python, Fortran, MatLab, and more can be used to read the data



Understanding a MODIS File Name

Level 2, 3 km, Aerosol Product

Product Name:

- Terra: MOD04
- Aqua: MYD04



HDFLook, Panoply, IDL, Python, Fortran, MatLab, and more can be used to read the data



MODIS Aerosol Parameters (SDS)

- Optical_Depth_Land_and_Ocean
 - Retrieved using Dark Target Algorithm
 - Only high quality data
 - Over land QA = 3
 - Over ocean QA = 1, 2, 3
 - 10 km and 3km
- Dark_Target_Deep_Blue_Optical_Depth_550_Combined
 - Deep Blue & Dark Target Algorithm Merged Product
 - 10 km only
- Quality_Assurance_Land
 - Quality flag associated with DD product



Quality Assurance is Extremely Important

QA indicates confidence in the quality of the retrieval

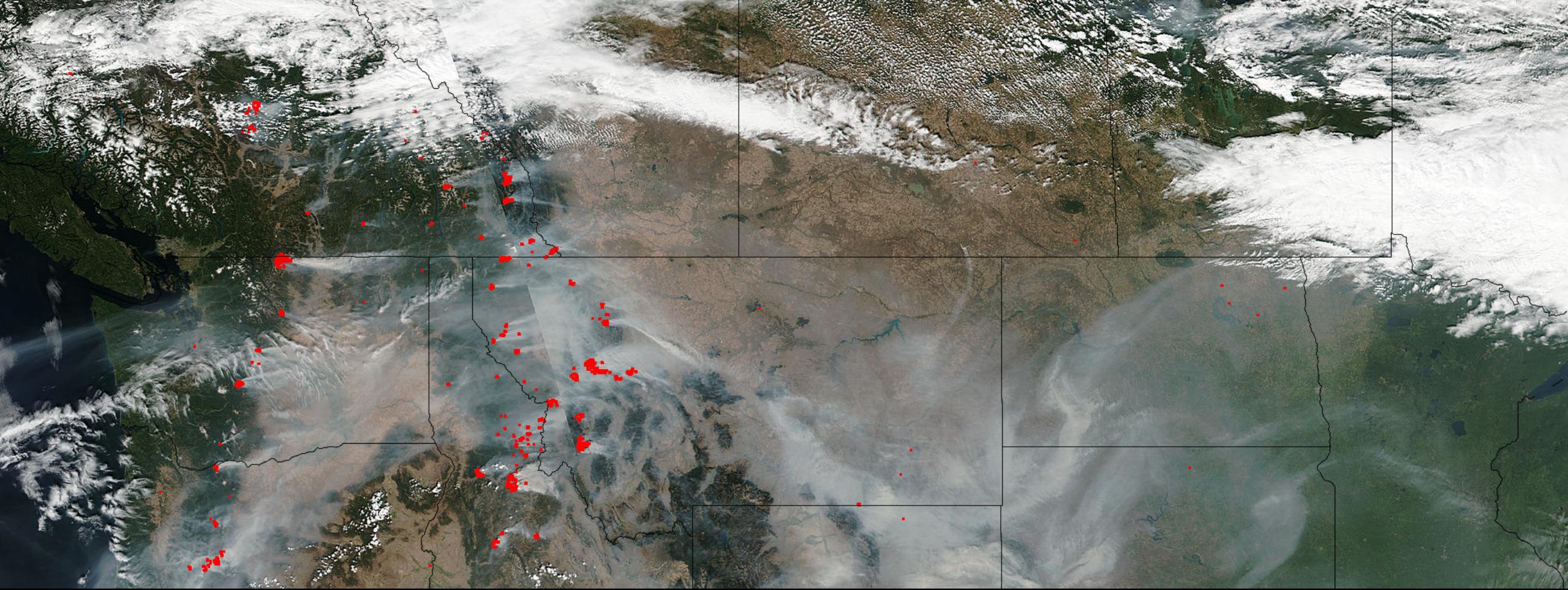
Quality_Assurance_Ocean

- Scale is 0 – 3
- Recommended Ocean QA above 1, 2, 3
- Factors:
 - number of pixels
 - error fitting
 - **how close to glint**

Quality_Assurance_Land

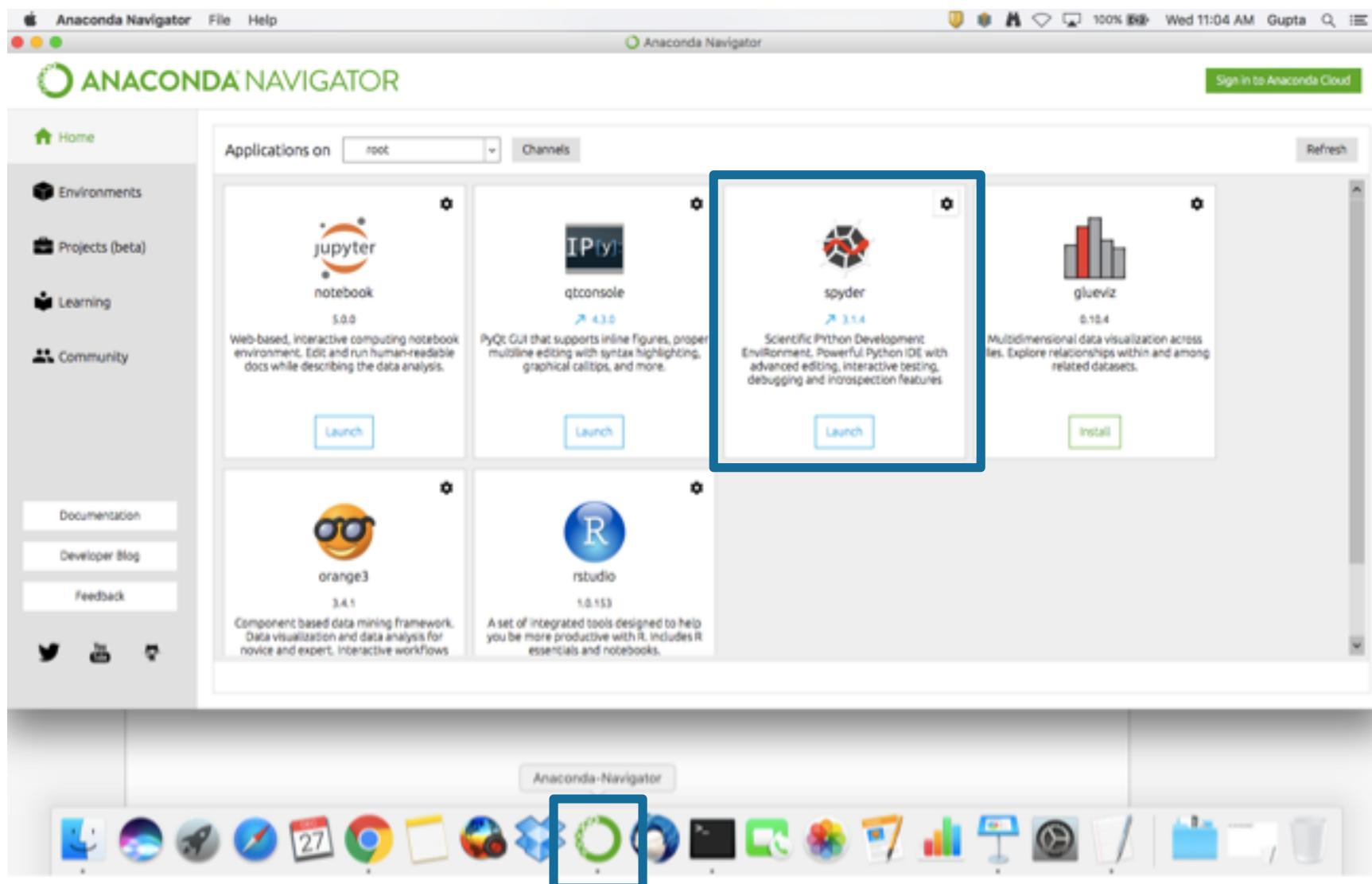
- Scale is 0 – 3
- Recommended Land QA of 3
- Factors:
 - number of pixels
 - error fitting
 - **surface reflectance**



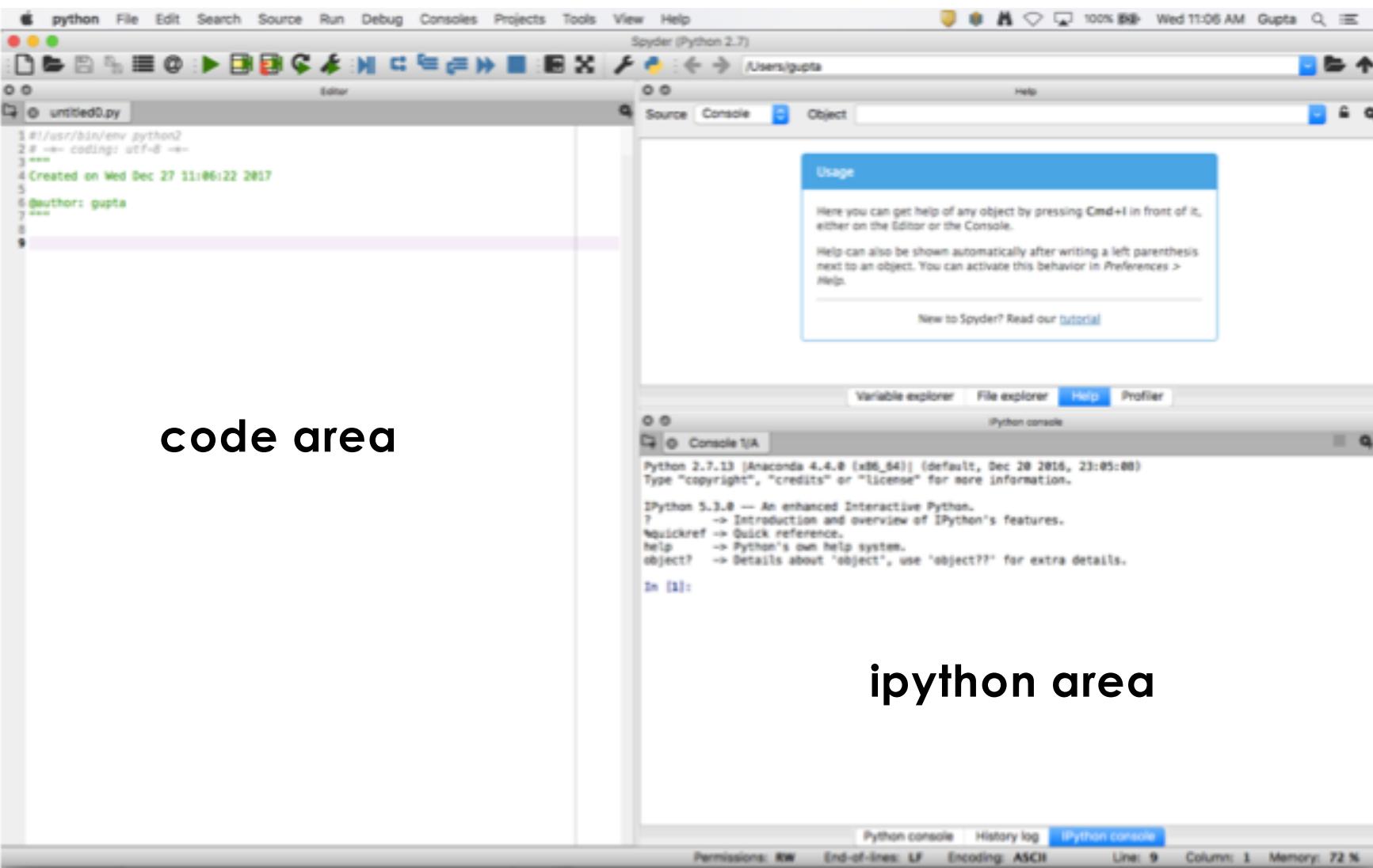


Getting Ready with Python

Anaconda & the Spyder Editor



Spyder View

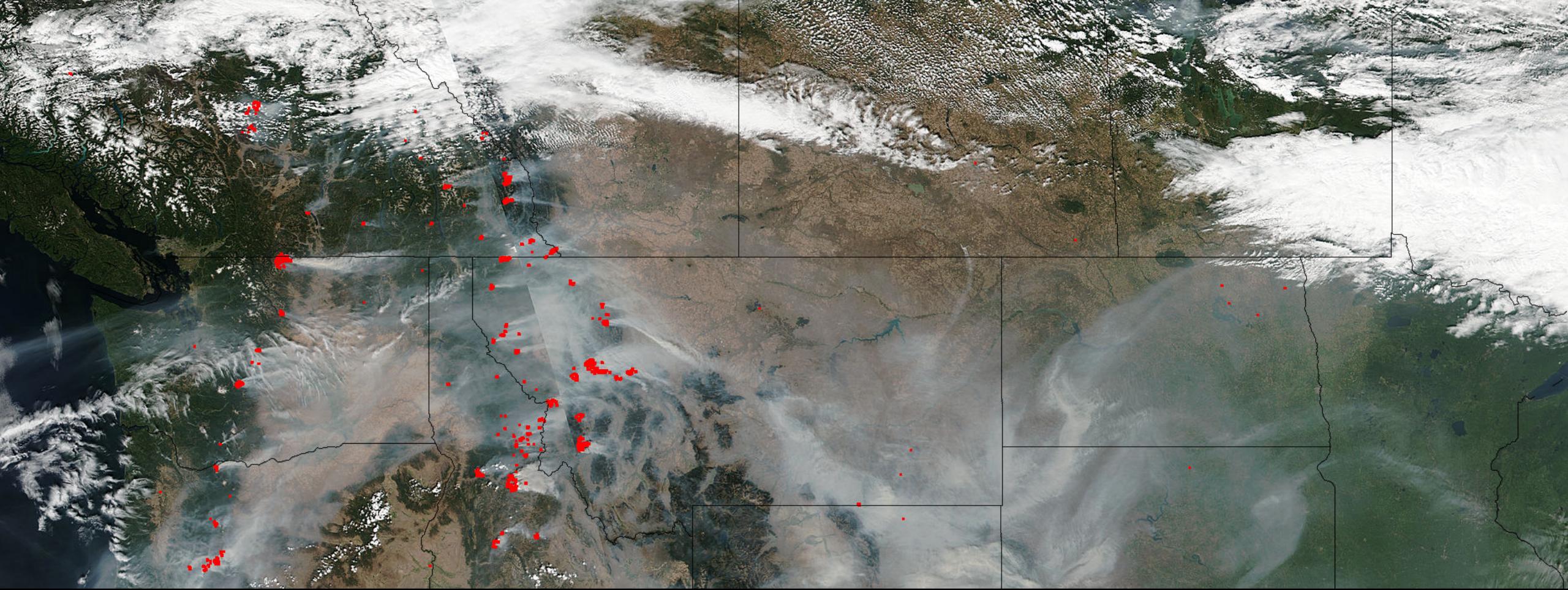


Current Directory View & File List

- Create a list of HDF files 'fileList.txt'
- The directory should have
 - All the python codes
 - All the HDF data files
 - A list of HDF files named as 'fileList.txt'

```
gs614-guptaml:CA_TRN gupta$ vi fileList.txt
gs614-guptaml:CA_TRN gupta$ ls
MYD04_3K.A2017232.2200.006.2017233154505.png
MYD04_3K.A2017232.2200.006.2017233154505.txt
MYD04_L2.A2017232.1520.006.2017233154749.png
MYD04_L2.A2017232.2200.006.2017233154546.png
MYD04_L2.A2017249.1925.006.2017250160408.hdf
MYD04_L2.A2017249.1925.006.2017250160408.txt
MYD04_L2.A2017249.1930.006.2017250160703.hdf
MYD04_L2.A2017249.2105.006.2017250160535.hdf
fileList.txt
py1
read_and_map_mod_aerosol.py
read_aod_and_calculate_pm25.py
read_mod_aerosol_and_dump_ascii.py
read_mod_aerosol_and_list_sds.py
read_mod_aerosol_at_a_location.py
readme
gs614-guptaml:CA_TRN gupta$ ls *.hdf
MYD04_L2.A2017249.1925.006.2017250160408.hdf
MYD04_L2.A2017249.1930.006.2017250160703.hdf
MYD04_L2.A2017249.2105.006.2017250160535.hdf
gs614-guptaml:CA_TRN gupta$ ls *.hdf >fileList.txt
gs614-guptaml:CA_TRN gupta$ more fileList.txt
MYD04_L2.A2017249.1925.006.2017250160408.hdf
MYD04_L2.A2017249.1930.006.2017250160703.hdf
MYD04_L2.A2017249.2105.006.2017250160535.hdf
gs614-guptaml:CA_TRN gupta$
```





Read a MODIS Aerosol File (HDF)
and Print SDS List

Print Scientific Data Sets (SDSs)

read_mod_aerosol_and_list_sds.py

- **Purpose:** read a MODIS aerosol level 2 data file in HDF format and print all the **Scientific Data Sets** (SDS)
- The code works for both 10 km and 3 km products

The screenshot shows the Spyder Python IDE interface. The top menu bar includes python, File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, Help. The title bar says Spyder (Python 2.7). The main area has tabs for Editor, Source, Console, and Object. The Editor tab shows the source code for `read_mod_aerosol_and_list_sds.py`. The code reads a file list, processes files based on their name (3km or 10km), and prints datasets from an HDF file. The Console tab shows the execution of the script and user interaction. The Python console output shows the script running and asking if it should process a specific file.

```
1 /**
2  * Modules: read_mod_aerosol_and_list_sds.py
3  * -----
4  * Disclaimer: The code is for demonstration purposes only. Users are responsible to check
5  *             the data quality and accuracy.
6  * Authors: Justin Roberts-Pierrel, 2015
7  * Organization: NASA ARSET
8  * Purpose: To print all SDS from an HDF4 file
9
10 See the README associated with this module for more information.
11 -----
12
13 #Import necessary modules
14 import pyhdf
15 import numpy as np
16
17 #This uses the file "fileList.txt", containing the list of files, in order to read them
18 try:
19     fileList=open('fileList.txt','r')
20 except:
21     print("Did not find a text file containing file names (perhaps name does not match")
22     sys.exit()
23
24 #Loops through all files listed in the text file
25 for FILE_NAME in fileList:
26     FILE_NAME=FILE_NAME.strip()
27     user_input=input('Would you like to process?' + FILE_NAME + '\n(Y/N)')
28     if user_input == 'N' or user_input == 'n':
29         continue
30     else:
31         if '3K' in FILE_NAME: #then this is a 3km MODIS file
32             print('\nThis is a MODIS 3km file. Here is a list of SDS in your file:\n')
33         elif 'L2' in FILE_NAME:
34             print('\nThis is a MODIS 10km file. Here is a list of SDS in your file:\n')
35         else:
36             print('The file named ' + FILE_NAME + ' is not a valid MODIS file (Or is it?)')
37             continue
38     try:
39         # open the hdf file for reading
40         hdfSD=SD(FILE_NAME)
41     except:
42         print('Unable to open file: ' + FILE_NAME + '\n Skipping...')
43         continue
44     #Extract the list of SDS in the hdf4 file
45     datasets=hdf.datasets()
46     #Print the list
47     for i,v in enumerate(datasets):
48         print('{0}. {1}'.format(i+1,v))
49     print ''
50     #asks if the user would like to continue to the next file, exits if not
51     user_input=input('Would you like to process the next file? (Y/N)') == 'Y'
52     print("\nAll valid files given have been processed")
53
```

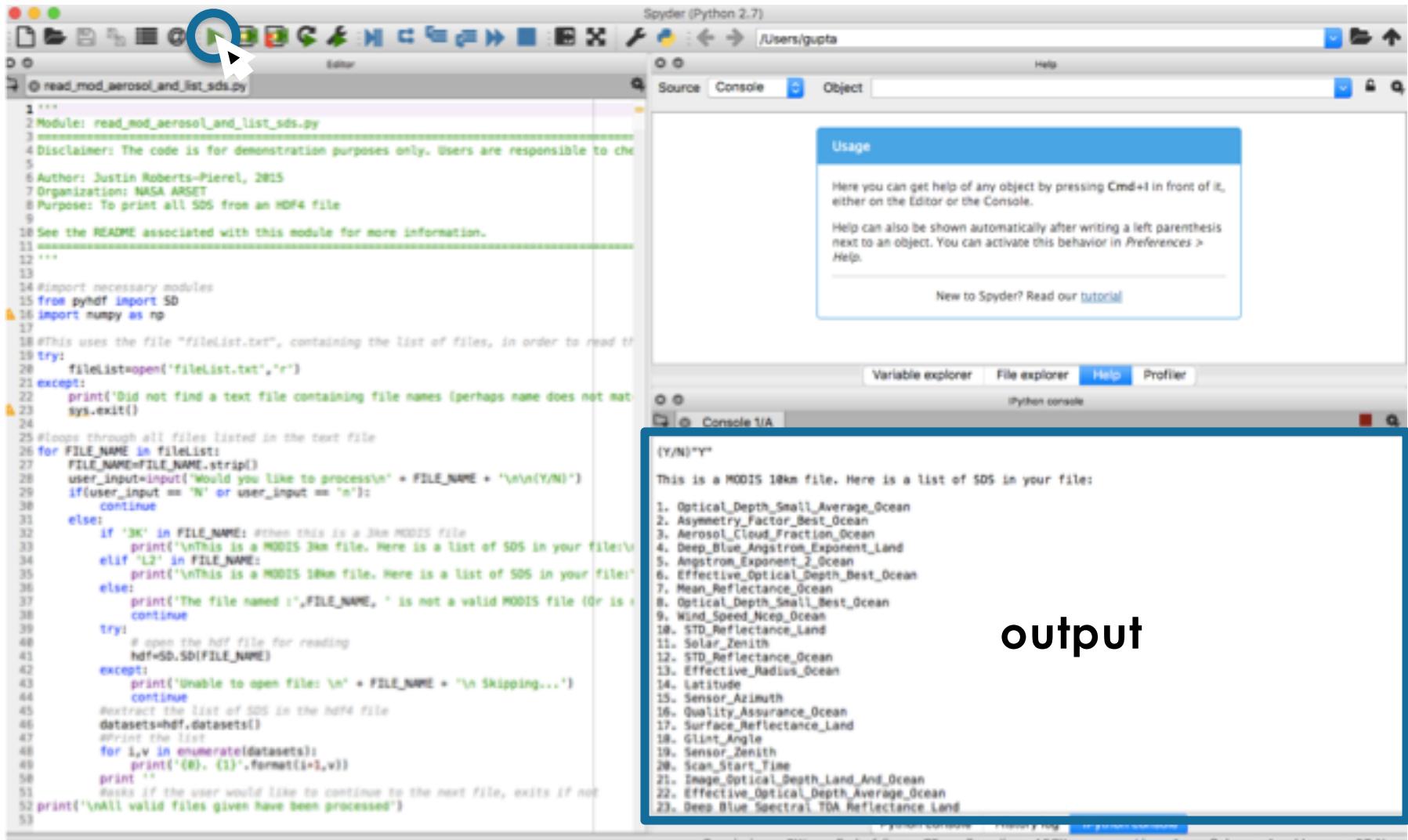
In [1]: runfile('/Users/gupta/Desktop/CA_TRN/read_mod_aerosol_and_list_sds.py', wdir='/Users/gupta/Desktop/CA_TRN')

Would you like to process
HY084_L2.A2017249.1925.006.2017250166408.hdf
(Y/N)?Y



Running and Output

- Click the green arrow to run the code
- The code will process all the files in the **fileList.txt** one-by-one
- Follow the instructions in the **ipython** terminal (i.e. enter ‘Y’ or ‘N’ when prompted and hit enter)



The screenshot shows the Spyder Python IDE interface. A green arrow icon is overlaid on the play button in the toolbar, indicating where to click to run the code. The code editor displays a script named `read_mod_aerosol_and_list_sds.py`. The console window shows the output of the script, which processes MODIS files and lists their SDS datasets. The word "output" is overlaid on the right side of the console window.

```
1 /**
2 Module: read_mod_aerosol_and_list_sds.py
3 -----
4 Disclaimer: The code is for demonstration purposes only. Users are responsible to che
5
6 Author: Justin Roberts-Pierrel, 2015
7 Organization: NASA ARSET
8 Purpose: To print all SDS from an HDF4 file
9
10 See the README associated with this module for more information.
11 -----
12
13
14 #Import necessary modules
15 from pyhdf import SD
16 import numpy as np
17
18 #This uses the file "fileList.txt", containing the list of files, in order to read th
19 try:
20     fileList=open('fileList.txt','r')
21 except:
22     print('Did not find a text file containing file names (perhaps name does not mat
23     sys.exit()
24
25 #Loops through all files listed in the text file
26 for FILE_NAME in fileList:
27     FILE_NAME=FILE_NAME.strip()
28     user_input=input('Would you like to process\n' + FILE_NAME + '\n(Y/N)')
29     if(user_input == 'N' or user_input == 'n'):
30         continue
31     else:
32         if '3K' in FILE_NAME: print('This is a MODIS 3km file')
33         elif 'L2' in FILE_NAME: print('This is a MODIS 1km file. Here is a list of SDS in your fil
34         else: print('This is a MODIS 18km file. Here is a list of SDS in your fil
35
36     print('The file named :',FILE_NAME, " is not a valid MODIS file (Or is it? )")
37     continue
38
39     try:
40         # open the hdf file for reading
41         hdf=SD(FILE_NAME)
42     except:
43         print('Unable to open file: \n' + FILE_NAME + '\n Skipping...')
44         continue
45     #Extract the list of SDS in the hdf file
46     datasets=hdf.datasets()
47     print the list
48     for i,v in enumerate(datasets):
49         print('{0}. {1}'.format(i+1,v))
50     print ''
51     #asks if the user would like to continue to the next file, exits if not
52 print('\nAll valid files given have been processed')
53
```

(Y/N)"Y"
This is a MODIS 18km file. Here is a list of SDS in your file:
1. Optical_Depth_Small_Average_Ocean
2. Asymmetry_Factor_Best_Ocean
3. Aerosol_Cloud_Fraction_Ocean
4. Deep_Blue_Angstrom_Exponent_Land
5. Angstrom_Exponent_J_Ocean
6. Effective_Optical_Depth_Best_Ocean
7. Mean_Reflectance_Ocean
8. Optical_Depth_Small_Best_Ocean
9. Wind_Speed_Ncep_Ocean
10. STD_Reflectance_Land
11. Solar_Zenith
12. STD_Reflectance_Ocean
13. Effective_Radius_Ocean
14. Latitude
15. Sensor_Azimuth
16. Quality_Assurance_Ocean
17. Surface_Reflectance_Land
18. Glint_Angle
19. Sensor_Zenith
20. Scan_Start_Time
21. Image_Optical_Depth_Land_And_Ocean
22. Effective_Optical_Depth_Average_Ocean
23. Deep_Blue_Spectral_TDA_Reflectance_Land



Editing the Code

```
Module: read_mod_aerosol_and_list_sds.py
=====
Disclaimer: The code is for demonstration purposes only. Users are responsible to check for accuracy and revise to fit their obj
Author: Justin Roberts-Pierel, 2015
Organization: NASA ARSET
Purpose: To print all SDS from an HDF4 file

See the README associated with this module for more information.
=====

#Import necessary modules
from pyhdf import SD
import numpy as np

#This uses the file "fileList.txt", containing the names of files, in order to read the files
try:
    fileList=open('fileList.txt',
except:
    print('Did not find a text file containing file names (perhaps name does not match)')
    sys.exit()

#loops through all files listed in the text file
for FILE_NAME in fileList:
    FILE_NAME=FILE_NAME.strip()
    user_input=input('Would you like to process\n' + FILE_NAME + '\n\n(Y/N)')
    if user_input == 'N' or user_input == 'n':
        continue
    else:
        if '3K' in FILE_NAME: #then this is a 3km MODIS file
            print('\nThis is a MODIS 3km file. Here is a list of SDS in your file:\n')
        elif 'L2' in FILE_NAME:
            print('\nThis is a MODIS 10km file. Here is a list of SDS in your file:\n')
        else:
            print('The file named :',FILE_NAME, ' is not a valid MODIS file (Or is named incorrectly). \n')
            continue
    try:
        # open the hdf file for reading
        hdf=SD.SD(FILE_NAME)
    except:
        print('Unable to open file: \n' + FILE_NAME + '\n Skipping...')
        continue
    #extract the list of SDS in the hdf4 file
    datasets=hdf.datasets()
    #Print the list
    for i,v in enumerate(datasets):
        print('{0}. {1}'.format(i+1,v))
    print ''
    #asks if the user would like to continue to the next file, exits if not
    print('\nAll valid files given have been processed')
```

change the name
of fileList.txt to any
name you'd like

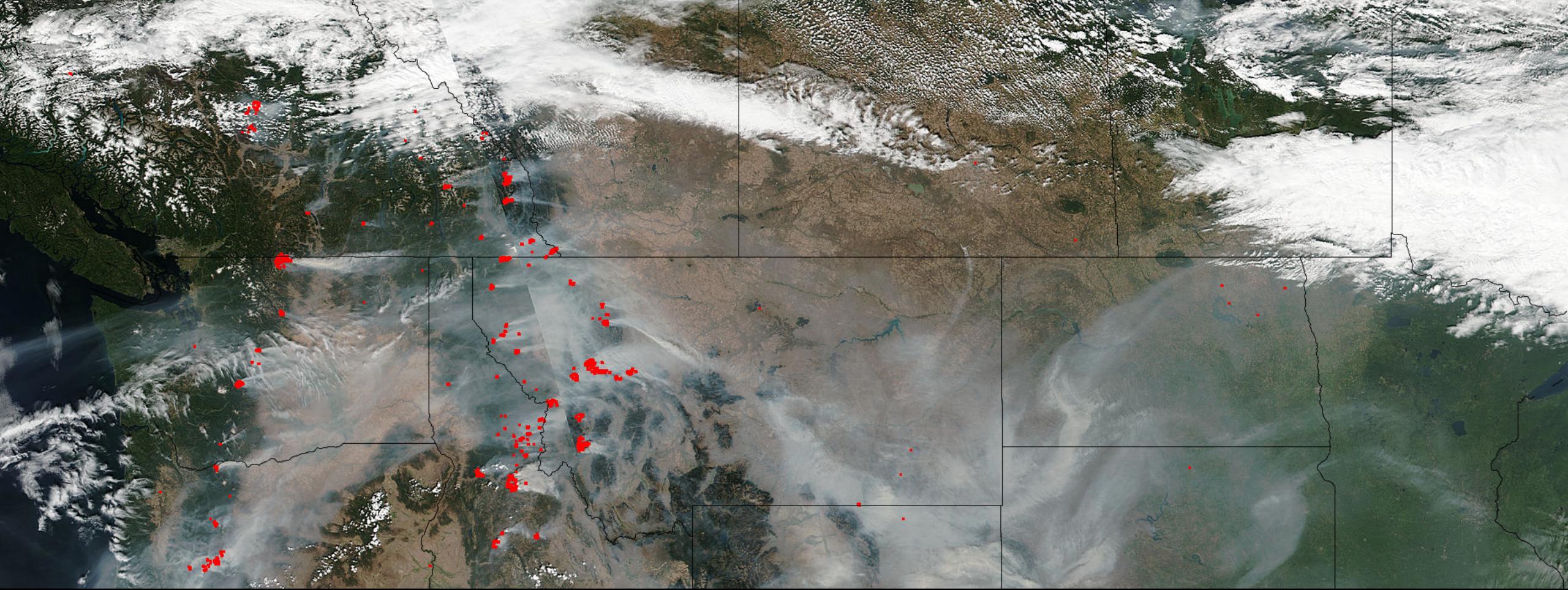
This code has
been tested for 3
km and 10 km
MODIS aerosol
Level 2 data files



Applications

- MODIS Level 2 aerosol data are provided in HDF files
- Each HDF file contains several geophysical parameters
- Special codes and tools are required to open HDF files
- This code helps users see the name of the available SDSs inside an HDF file for further analysis





Map Aerosol Optical Depth

Plot and save a map of MODIS AOD

read_and_map_mod_aerosol.py

The screenshot shows the Spyder Python IDE interface. On the left, the code editor displays the script `read_and_map_mod_aerosol.py`. The script reads a file list from `fileList.txt`, processes user input to determine if a MODIS file is 3km or 10km, and then tries to open the file. If successful, it retrieves latitude and longitude information. It then tries to select a specific SDS (Scientific Data Set) named `SDS_NAME`. The right side of the interface includes a help browser titled "Usage" and an IPython console window showing the Python environment and some basic help commands.

```
Spyder (Python 2.7)
Editor
Source Console Object
Help

read_and_map_mod_aerosol.py

13 #!/usr/bin/python
14
15 #import necessary modules
16 from pyhdf import SD
17 import numpy as np
18 from mpl_toolkits.basemap import Basemap
19 import matplotlib.pyplot as plt
20 import sys
21
22 #this uses the file "fileList.txt", containing the list of files, in order to read the files
23 try:
24     fileList=open('fileList.txt','r')
25 except:
26     print('Did not find a text file containing file names (perhaps name does not match!)')
27     sys.exit()
28
29 #loop through all files listed in the text file
30 for FILE_NAME in fileList:
31     FILE_NAME=FILE_NAME.strip()
32     user_input=input('Would you like to process\n' + FILE_NAME + '\n\n(Y/N)')
33     if user_input == 'N' or user_input == 'n':
34         continue
35     else:
36         if '3K' in FILE_NAME: #then this is a 3km MODIS file
37             print('This is a 3km MODIS file. Here is some information: ')
38             SDS_NAME='Optical_Depth_Land_And_Ocean' # The name of the sds to read
39         elif '10' in FILE_NAME: # same as above but for 10km MODIS file
40             print('This is a 10km MODIS file. Here is some information: ')
41             SDS_NAME='AOD_550_Dark_Target_Deep_Blue_Combined'
42         else: #if it is neither 3km nor 10km, then this will skip the rest of this loop iteration
43             print('The file :',FILE_NAME, ' is not a valid MODIS file (or is named incorrectly). We')
44             continue
45         try:
46             # open the hdf file for reading
47             hdf=SD(FILE_NAME)
48         except:
49             print('Unable to open file: \n' + FILE_NAME + "\n Skipping...")
50             continue
51
52         # get lat and lon info
53         lat = hdf.select('Latitude')
54         latitude = lat[:]
55         min_lat=latitude.min()
56         max_lat=latitude.max()
57         lon = hdf.select('Longitude')
58         longitude = lon[:]
59         min_lon=longitude.min()
60         max_lon=longitude.max()
61
62         #get AOD SDS, or exit if it doesn't find the SDS in the file
63         try:
64             SDS=hdf.select( SDS_NAME )
65         except:
66             print('Sorry.. your MODIS hdf file does not contain the SDS! ', SDS_NAME, '. Please try another file')

Run file Permissions: RW End-of-lines: CR Encoding: ASCII Line: 51 Column: 3 Memory: 67 %
Python console History log IPython console
```



Running and Output

AOD statistics

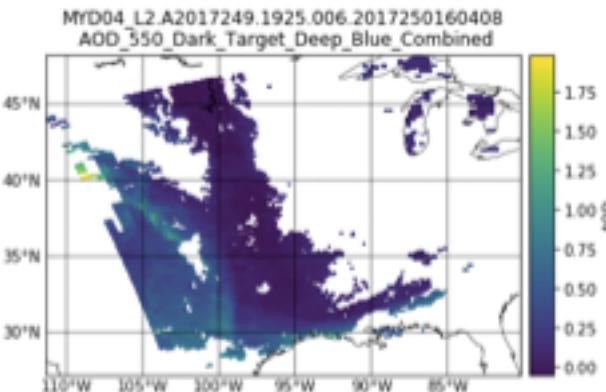
```
In [1]: runfile('/Users/gupta/Desktop/CA_TRN/read_and_map_mod_aerosol.py', wdir='/Users/gupta/Desktop/CA_TRN')
```

```
Would you like to process  
MYD04_L2.A2017249.1925.006.2017250160408.hdf
```

```
(Y/N)"Y"  
This is a 10km MODIS file. Here is some information:  
('The valid range of values is: ', -0.1, ' to ', 5.0, '\nThe average is: ', 0.178, '\nThe standard deviation is: ', 0.23)  
('The range of latitude in this file is: ', 27.187273, ' to ', 48.299458, 'degrees \nThe range of longitude in this file is: ', -111.39777, ' to ', -80.255447, ' degrees')
```

```
Would you like to create a map of this data? Please enter Y or N
```

```
"Y"  
/Users/pgupta3/python/anaconda/lib/python2.7/site-packages/mpl_toolkits/basemap/__init__.py:3413:  
MatplotlibDeprecationWarning: The ishold function was deprecated in version 2.0.  
    b = ax.ishold()  
/Users/pgupta3/python/anaconda/lib/python2.7/site-packages/mpl_toolkits/basemap/__init__.py:3422:  
MatplotlibDeprecationWarning: axes.hold is deprecated.  
    See the API Changes document (http://matplotlib.org/api/api\_changes.html)  
    for more details.  
    ax.hold(b)
```



Output AOD map

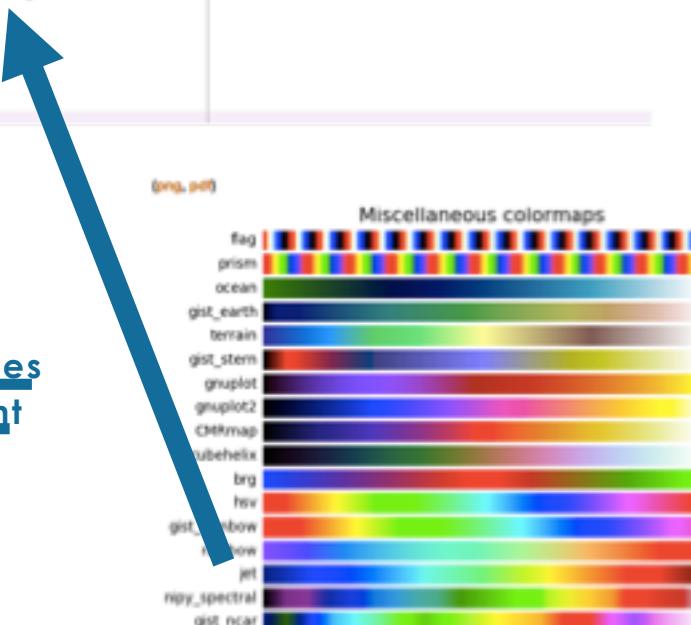
```
Would you like to save this map? Please enter Y or N
```



Editing the Code

Change the Color Scale

```
data=data.astype('float')
data[data == fv] = np.nan
#create the map
data = np.ma.masked_array(data, np.isnan(data))
m = Basemap(projection='cyl', resolution='l', llcrnrlat=min_lat, urcrnrlat = max_lat, llcrnrlon=min_lon,
m.drawcoastlines(linewidth=0.5)
m.drawparallels(np.arange(-90., 120., 5.), labels=[1, 0, 0, 0])
m.drawmeridians(np.arange(-180., 180., 5.), labels=[0, 0, 0, 1])
x, y = m(longitude, latitude)
m.pcolormesh(x, y, data*scale_factor, cmap=plt.cm.jet)
plt.autoscale()
#create colorbar
cb = m.colorbar()
#label colorbar
cb.set_label('AOD')
```



https://matplotlib.org/examples/color/colormaps_reference.html

Change the SDS

```
#loops through all files listed in the text file
for FILE_NAME in fileList:
    FILE_NAME=FILE_NAME.strip()
    user_input=input("\nWould you like to process\n" + FILE_NAME + "\n\n(Y/N)")
    if(user_input == 'N' or user_input == 'n'):
        continue
    else:
        if '3K' in FILE_NAME:#then this is a 3km MODIS file
            print('This is a 3km MODIS file. Here is some information: ')
            SDS_NAME='Optical_Depth_Land_And_Ocean' # The name of the sds to read
        elif 'L2' in FILE_NAME: #Same as above but for 10km MODIS file
            print('This is a 10km MODIS file. Here is some information: ')
            SDS_NAME='AOD_558_Dark_Target_Deep_Blue_Combined'
        else:#if it is neither 3km nor 10km, then this will skip the rest of this loop iteration
            print('The file :',FILE_NAME, ' is not a valid MODIS file (Or is named incorrectly). \n')
            continue
    try:
        # open the hdf file for reading
        hdf=SD.SD(FILE_NAME)
    except:
        print('Unable to open file: \n' + FILE_NAME + '\n Skipping... ')
        continue
```

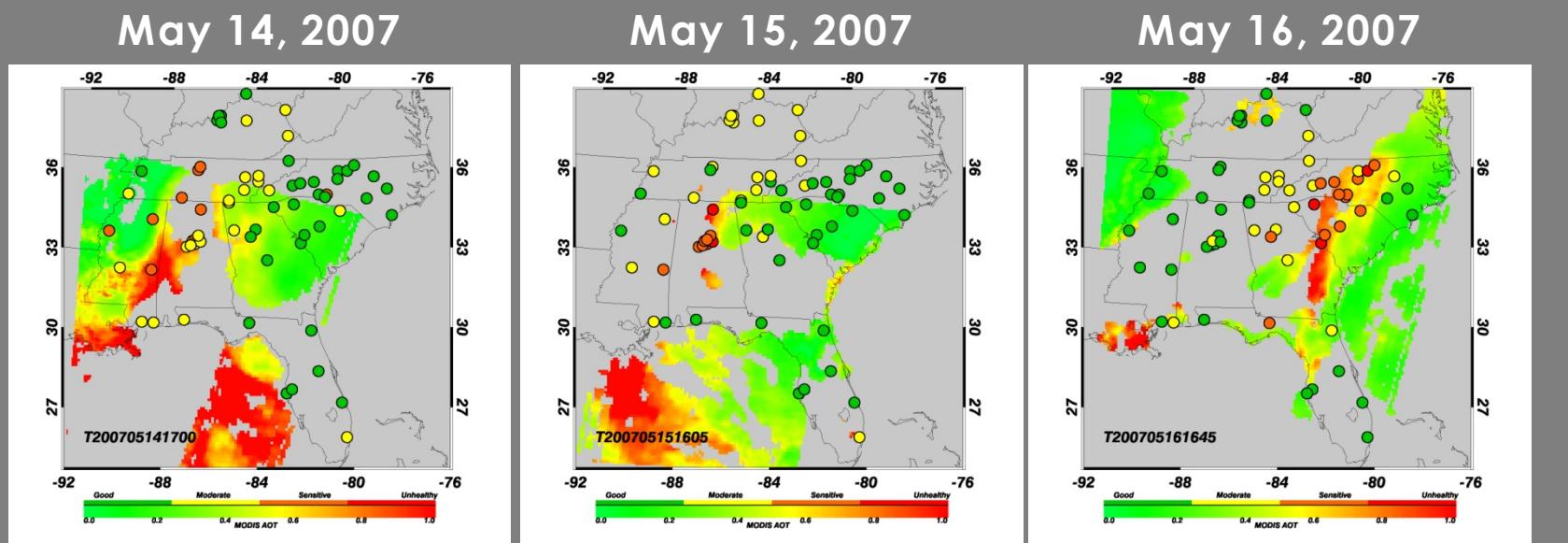


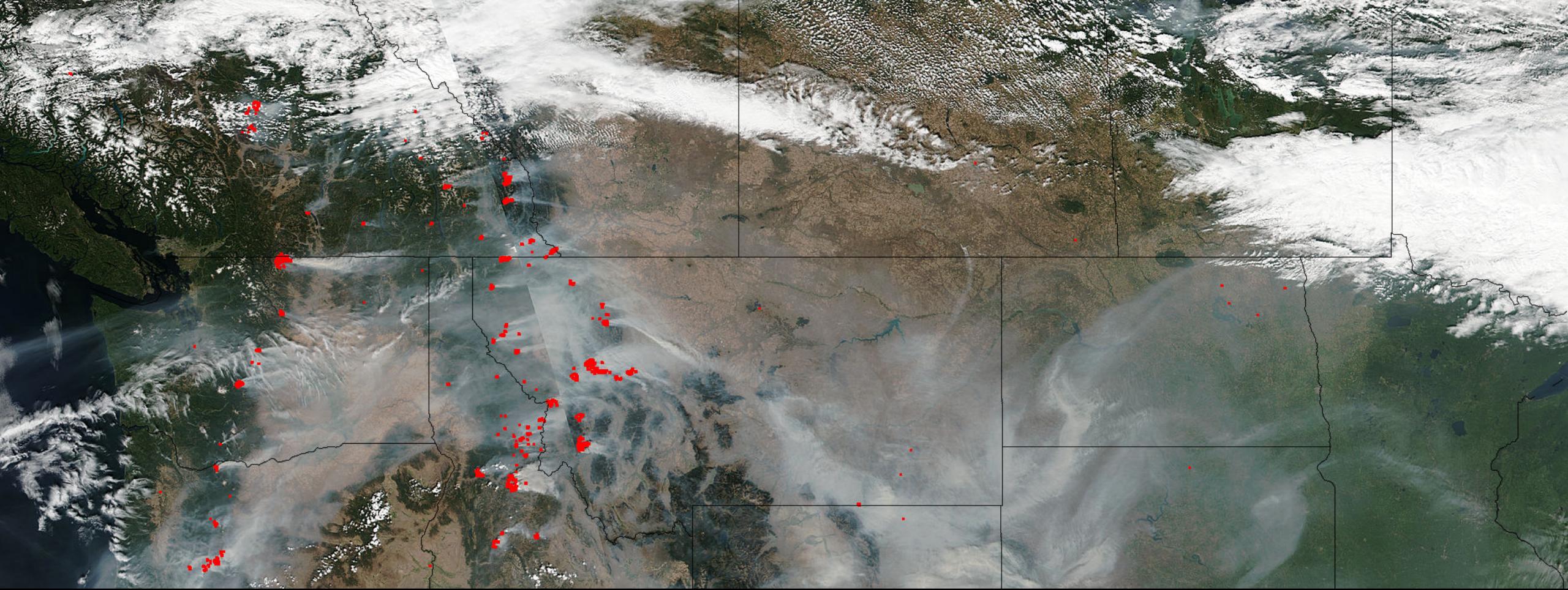
Applications

- This is a sample code to read and map the MODIS Level 2 aerosol data
- The code can be modified to address different mapping needs
- Users can create daily maps of AOD over certain regions and start analyzing changes over time
- AOD maps can also help identify regions with high pollution levels

Example:

High AOD values from smoke show good agreement with surface monitors (circles).





Extract AOD at a Surface Station

Extract AOD Values at a given location

read_mod_aerosol_at_a_location.py

- **Purpose:** read a MODIS aerosol level 2 data file in HDF format and extract AOD values at a given ground location
- The code works for both 10 km and 3 km products

The screenshot shows the Spyder Python IDE interface. The left pane displays the code for `read_mod_aerosol_at_a_location.py`. The right pane shows the execution of the code in the "Console 3/A" tab. The user enters coordinates (48.5, -98.5) and specifies a 3x3 grid. The code processes the MODIS file and prints the AOD values for the specified pixel and its neighborhood.

```
#!/usr/bin/python
...
#Module: read_mod_aerosol_at_a_location.py
#-----
#Disclaimer: The code is for demonstration purposes only. Users are responsible to check for accuracy and relevance.
#Author: Justin Roberts-Pierrel, 2013
#Organization: NASA ARSET
#Purpose: To view info about a variety of SDS from a MODIS HDF4 file (or series of files) both generally and specifically
#See the README associated with this module for more information.
...
#Import necessary modules
from pyhdf import SD
import numpy as np
...
#This uses the file "fileList.txt", containing the list of files, in order to read the files
try:
    fileList=open('fileList.txt','r')
except:
    print("Did not find a text file containing file names (perhaps name does not match?)")
    sys.exit()
...
#Loop through all files listed in the text file
for FILE_NAME in fileList:
    FILE_NAME=FILE_NAME.strip()
    user_input=input("Would you like to process? " + FILE_NAME + "\n(Y/N)?")
    if(user_input == 'Y' or user_input == 'y'):
        continue
    else:
        if '3K' in FILE_NAME: #When this is a 3km MODIS file
            user_input=int(input("Which SDS would you like to view? (Type the number and press enter) \n:"))
            while user_input not in (1,2,3,4): #Respects the question if the user does not choose one of the options
                print("Please try again.")
                user_input=int(input("Which SDS would you like to view? (Type the number and press enter) \n:"))
            dataFields=sdct[1,'Optical_Depth_Land_And_Ocean'],(2,'Land_Ocean_Quality_Flag'),(3,'Image_Type'),(4,'Cloud_Quality')
            if 'L2' in FILE_NAME: #Same as above but for 3km MODIS file
                user_input=int(input("Which SDS would you like to view? (Type the number and press enter) \n:"))
                while user_input not in (1,2,3):
                    print("Please try again.")
                    user_input=int(input("Which SDS would you like to view? (Type the number and press enter) \n:"))
            dataFields=sdct[(1,'Deep_Blue_Aerosol_Optical_Depth_558_Land'),(2,'AOD_558_Dark_Target_Deep_Blue_Combined'),(3,'AOD_558_Dark_Target_Deep_Blue_Combined_OA_Flag'),(4,'Cloud_Quality')]
            SDS_NAME=dataFields[int(userInput)] #The name of the SDS to read
        try:
            #Open the hdf file for reading
            hdf=SD(FILE_NAME)
        except:
            print("Unable to open file: " + FILE_NAME + "\n Skipping...")
            continue
        ...
degrees?
Please enter the latitude you would like to analyze (Deg. N): 48.5
Please enter the longitude you would like to analyze (Deg. E): -98.5
('InThe nearest pixel to your entered location is at: \nLatitude: ', 48.030125, 'Longitude: ', -98.5548514)
('The value of ', 'AOD_558_Dark_Target_Deep_Blue_Combined', 'at this pixel is', '-9999L', '(No Value)\n')
There are no valid pixels in a 3x3 grid centered at your entered location.
There are no valid pixels in a 5x5 grid centered at your entered location.

Would you like to process
MOD04_L2.A2017249.1938.006.2817250160703.hdf
(Y/N)?Y
Which SDS would you like to view? (Type the number and press enter)
(1) Deep_Blue_Aerosol_Optical_Depth_558_Land
(2) AOD_558_Dark_Target_Deep_Blue_Combined
(3) AOD_558_Dark_Target_Deep_Blue_Combined_OA_Flag
(4) Cloud_Quality
1
('The range of latitude in this file is: ', 44.253548, ' to ', 48.211197, ' degrees'
'\nThe range of longitude in this file is: ', -126.93629, ' to ', -82.291889, ' degrees')

Please enter the latitude you would like to analyze (Deg. N): 49.5
Please enter the longitude you would like to analyze (Deg. E): -98.5
('InThe nearest pixel to your entered location is at: \nLatitude: ', 49.482555, 'Longitude: ', -98.51669)
('The value of ', 'Deep_Blue_Aerosol_Optical_Depth_558_Land', 'at this pixel is', 0.173)
('InThere', 'are', 9, 'valid', 'pixels', 'in a 3x3 grid centered at your entered location.')
('The average value in this grid is: ', 0.197, '\nThe median value in this grid is: ', 0.171, '\nThe standard deviation in this grid is: ', 0.075)
('InThere', 'are', 24, 'valid', 'pixels', 'in a 5x5 grid centered at your entered location.')
('The average value in this grid is: ', 0.204, '\nThe median value in this grid is: ', 0.195, '\nThe standard deviation in this grid is: ', 0.077)

Would you like to process
MOD04_L2.A2017249.2105.006.2817250160535.hdf
(Y/N)?
```



Running and Output

Type "Y" to process file,
"N" to skip

Select SDS

Lat & Lon of station

Outputs

x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x

```
Would you like to process  
MYD04_L2.A2017249.1930.006.2017250160703.hdf  
(Y/N)"Y"  
  
Which SDS would you like to view? (Type the number and press enter)  
(1) Deep_Blue_Aerosol_Optical_Depth_550_Land  
(2) AOD_550_Dark_Target_Deep_Blue_Combined  
(3) AOD_550_Dark_Target_Deep_Blue_Combined_QA_Flag  
1  
('The range of latitude in this file is: ', 44.253548, ' to ', 66.211197, 'degrees  
\nThe range of longitude in this file is: ', -126.93629, ' to ', -82.291809, '  
degrees')  
  
Please enter the latitude you would like to analyze (Deg. N): 49.5  
Please enter the longitude you would like to analyze (Deg. E): -100.5  
('\nThe nearest pixel to your entered location is at: \nLatitude:', 49.482555, '  
Longitude:', -100.51669)  
('The value of ', 'Deep_Blue_Aerosol_Optical_Depth_550_Land', 'at this pixel is ',  
0.171)  
('\nThere', 'are', 9, 'valid', 'pixels', 'in a 3x3 grid centered at your entered  
location.')  
('\nThe average value in this grid is: ', 0.197, '\nThe median value in this grid  
is: ', 0.171, '\nThe standard deviation in this grid is: ', 0.075)  
('\nThere', 'are', 24, 'valid', 'pixels', 'in a 5x5 grid centered at your entered  
location. \n')  
('The average value in this grid is: ', 0.204, '\nThe median value in this grid  
is: ', 0.201, '\nThe standard deviation in this grid is: ', 0.077)  
  
Would you like to process  
MYD04_L2.A2017249.2105.006.2017250160535.hdf  
(Y/N)
```



Editing the Code – Change the SDS

```
s through all files listed in the text file
FILE_NAME in fileList:
FILE_NAME=FILE_NAME.strip()
user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
if(user_input == 'N' or user_input == 'n'):
    continue
else:
    if '3K' in FILE_NAME: #then this is a 3km MODIS file
        userInput=int(input('Which SDS would you like to view? (Type the number and press enter) \n(1) Optical_Depth_Land_And_Ocean \n(2)
        while userInput not in {1,2,3,4}:#repeats the question if the user does not choose one of the options
            print('Please try again.')
            userInput=int(input('Which SDS would you like to view? (Type the number and press enter) \n(1) Optical_Depth_Land_And_Ocean
        #Uses a Python dictionary to choose the SDS indicated by the user
        dataFields=dict([(1,'Optical_Depth_Land_And_Ocean'),(2,'Land_Ocean_Quality_Flag'),(3,'Image_Optical_Depth_Land_And_Ocean'),(4,'L
    elif 'L2' in FILE_NAME:#Same as above but for 10km MODIS file
        userInput=int(input('Which SDS would you like to view? (Type the number and press enter) \n(1) Deep_Blue_Aerosol_Optical_Depth_5
        while userInput not in {1,2,3}:
            print('Please try again.')
            userInput=int(input('Which SDS would you like to view? (Type the number and press enter) \n(1) Deep_Blue_Aerosol_Optical_D
        dataFields=dict([(1,'Deep_Blue_Aerosol_Optical_Depth_550_Land'),(2,'AOD_550_Dark_Target_Deep_Blue_Combined'),(3,'AOD_550_Dark_Ta
    SDS_NAME=dataFields[int(userInput)] # The name of the sds to read
try:
    # open the hdf file for reading
    hdf=SD.SD(FILE_NAME)
except:
    print('Unable to open file: \n' + FILE_NAME + '\n Skipping...')
    continue
```



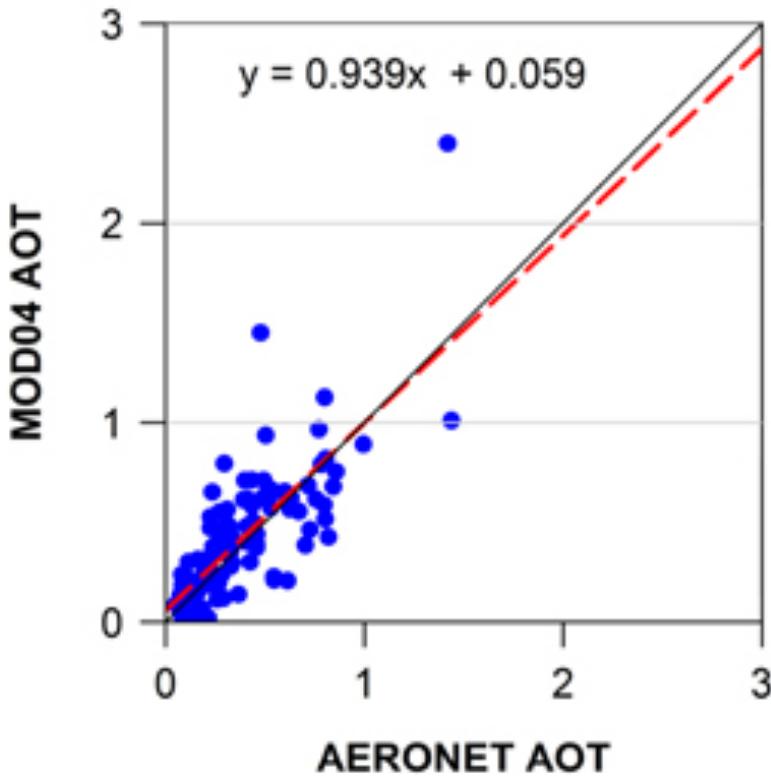
Editing the Code – Change the AOD Calculations

```
#calculates mean, median, stdev in a 3x3 grid around nearest point to entered location
if x < 1:
    x+=1
if x > data.shape[0]-2:
    x-=2
if y < 1:
    y+=1
if y > data.shape[1]-2:
    y-=2
three_by_three=data[x-1:x+2,y-1:y+2]
three_by_three=three_by_three.astype(float)
three_by_three[three_by_three==float(fillvalue)]=np.nan
nnan=np.count_nonzero(~np.isnan(three_by_three))
if nnan == 0:
    print ('\nThere are no valid pixels in a 3x3 grid centered at your entered location.')
else:
    three_by_three=three_by_three*scale_factor
    three_by_three_average=np.nanmean(three_by_three)
    three_by_three_std=np.nanstd(three_by_three)
    three_by_three_median=np.nanmedian(three_by_three)
    if nnan == 1:
        npixels='is'
        mpixels='pixel'
    else:
        npixels='are'
        mpixels='pixels'
print ('\nThere',npixels,nnan,'valid',mpixels,'in a 3x3 grid centered at your entered location.')
print ('\nThe average value in this grid is: ',round(three_by_three_average,3),' \nThe median value in this grid is: ',round(three,
```

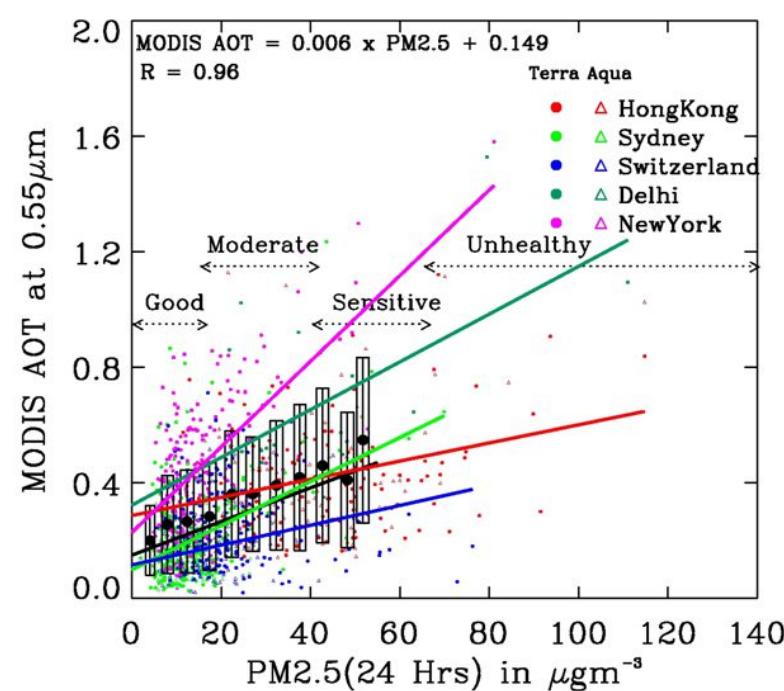


Applications

Satellite AOD Validation



AOD-PM_{2.5} Relationship



Time Series Analysis

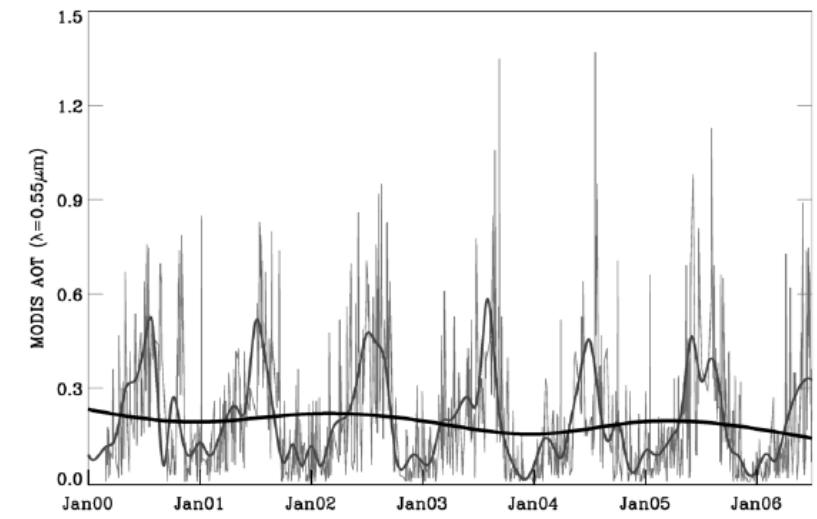
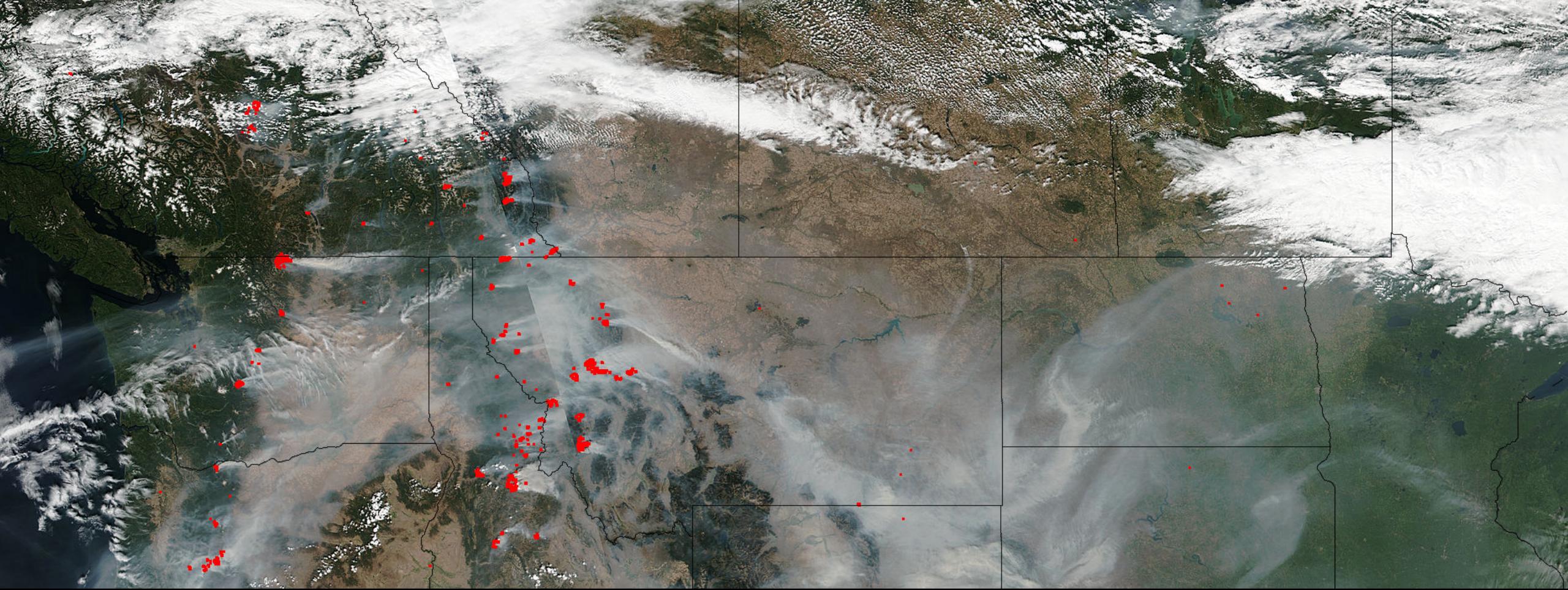


Image Sources: Gupta et al., Gupta et al., 2006, Gupta et al., 2007



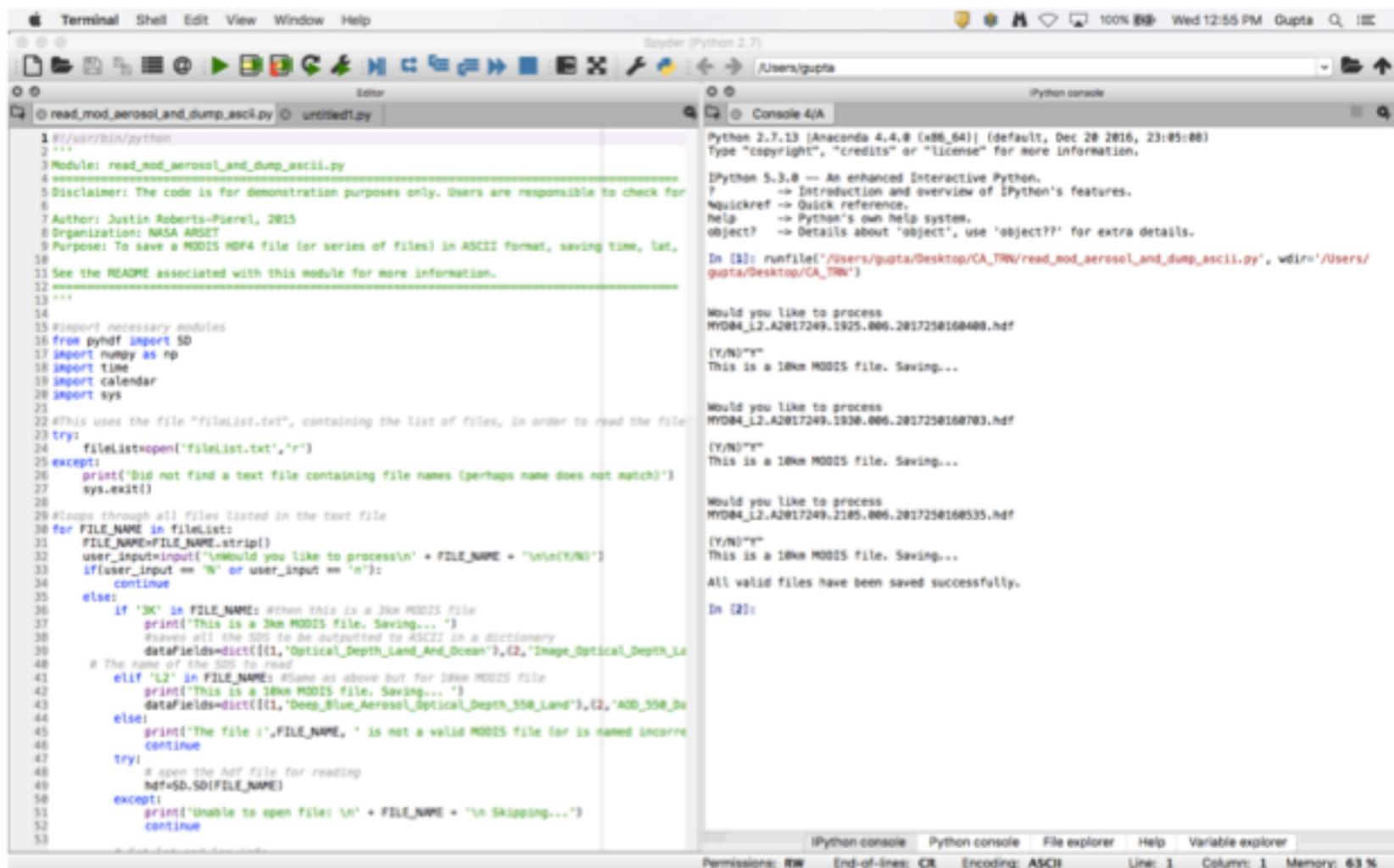


Output HDF Variables to CSV

Output MODIS Aerosol Level 2 HDF variables to a CSV file

read_mod_aerosol_and_dump_ascii.py

- **Purpose:** read a MODIS aerosol level 2 data file in HDF format and write certain SDSs into a csv (text) file
- The code works for both 10 km and 3 km products



The screenshot shows the Spyder Python IDE interface. On the left, the code editor displays the script `read_mod_aerosol_and_dump_ascii.py`. The script reads a list of files from `fileList.txt`, processes them, and saves them as ASCII files. It includes logic to handle 3km and 10km MODIS files. On the right, the IPython console shows the execution of the script. It prompts the user to process a 10km MODIS file, which it saves successfully. Then, it prompts for another file, also a 10km MODIS file, which it saves successfully. Finally, it prints a message stating all valid files have been saved successfully.

```
#!/usr/bin/python
...
#Module: read_mod_aerosol_and_dump_ascii.py
#-----
#Disclaimer: The code is for demonstration purposes only. Users are responsible to check for
#Author: Justin Roberts-Pieren, 2013
#Organization: NASA ARSET
#Purpose: To save a MODIS HDF4 file (or series of files) in ASCII format, saving time, lat,
#See the README associated with this module for more information.
...
#Import necessary modules
from pyhdf import SD
import numpy as np
import time
import calendar
import sys
...
#This uses the file "fileList.txt", containing the list of files, in order to read the file
try:
    fileList=open('fileList.txt','r')
except:
    print("Did not find a text file containing file names (perhaps name does not match?)")
    sys.exit()
...
for FILE_NAME in fileList:
    FILE_NAME=FILE_NAME.strip()
    user_input=input("Would you like to process\n'"+FILE_NAME+"\n(Y/N)?")
    if user_input == 'N' or user_input == 'n':
        continue
    else:
        if '3K' in FILE_NAME: #Then this is a 3km MODIS file
            print("This is a 3km MODIS file. Saving... ")
            #Saves all the SDS to be outputted to ASCII in a dictionary
            dataFields=dict([(1,'Optical_Depth_Land_Ocean'),(2,'Image_Optical_Depth_La
# The name of the SDS to read
            elif 'L2' in FILE_NAME: #Same as above but for 10km MODIS file
                print("This is a 10km MODIS file. Saving... ")
                dataFields=dict([(1,'Deep_Blue_Aerosol_Optical_Depth_558_Land'),(2,'AOD_558_5a
            else:
                print('The file '+FILE_NAME+' is not a valid MODIS file (or is named incorrect
                continue
            try:
                # open the hdf file for reading
                hdf=SD.SD(FILE_NAME)
            except:
                print('Unable to open file: '+FILE_NAME+'\n Skipping...')
                continue
...

```



Output

MYDD4_L2,A2017249,1925.006,2017250160408.txt ~

Year,Month,Day,Hour,Minute,Second,Latitude,Longitude,Deep Blue Aerosol Optical Depth 558_Land,AOD_558_Dark_Target_Deep_Blue_Combined,AOD_558_Dark_Target_Deep_Blue_Combined_OA_Flag

2017,0,9,0,6,0,19,0,25,0,9,0,30,4542312622,-88,2554473877,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,30,4285984839,-88,7235641479,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,30,4832482839,-81,1592487227,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,30,3782196845,-81,5666427812,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,30,3535698388,-81,9493826733,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,30,329334259,-82,3896888381,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,30,3855496216,-82,6497421265,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,30,2822836743,-82,9728386396,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,30,2592868885,-83,2782287598,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,30,2367897834,-83,569961772,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,30,2147369385,-83,8481216431,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,30,1938789948,-84,1143835889,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,30,1718177795,-84,3694152832,0,322888815294,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,30,1589361267,-84,6143875122,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,30,1384397583,-84,8497695923,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,30,1183876935,-85,8764887568,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,30,89851289514,-85,2949752888,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,30,8738792542,-85,87859127888,0,491488823559,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,30,8519447327,-85,7188372314,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,30,6331192817,-85,9876538886,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,30,8145874823,-86,8991973877,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,30,9963378986,-86,2858646973,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,9783554877,-86,4656219482,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,9686361389,-86,6411895752,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,9431618187,-86,8128889127,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,925924885,-86,9785919189,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,9889858293,-87,1489683228,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,8921831952,-87,2994613647,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,87558354,-87,4542999268,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,8599884344,-87,6856747437,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,8428764343,-87,7537918891,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,8268318176,-87,8988342285,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,8189512329,-88,8489774478,-9999.0,0,483888822941,1,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,7952278588,-88,1883741455,-9999.0,0,537888825586,3,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,7796516418,-88,3171693895,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,7642173767,-88,4515228273,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,7489128113,-88,5835418781,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,7337341309,-88,7133789862,-9999.0,0,523888824841,1,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,7186717987,-88,8411333177,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,7837162781,-88,9669265747,-9999.0,0,488888819999,1,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,688867569,-89,9988668889,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,6741184126,-89,2138584717,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,6594429816,-89,3335876465,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,6448535919,-89,452545266,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,6303424835,-89,5708378418,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,6158943176,-89,6861343384,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,6815898942,-89,8889262885,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,587179184,-89,9144897441,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,5728931427,-90,8269812451,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,5586598785,-90,1382369995,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,5444612231,-90,248558415,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,53826119934,-90,3579177856,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,5161894666,-90,4653772583,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,5819721985,-90,5748288151,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,4878482819,-90,6888853149,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,4737319946,-90,7878483398,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,4596118927,-90,8925476874,-9999.0,-9999.0,0,0,0
2017,0,9,0,6,0,19,0,25,0,9,0,29,4474641614,-90,9974562114,-9999.0,-9999.0,0,0,0

This code saves a .csv file, which can be opened by excel, a text editor, or other codes or software



Editing the Code

Change the list
SDS to be written
as an output

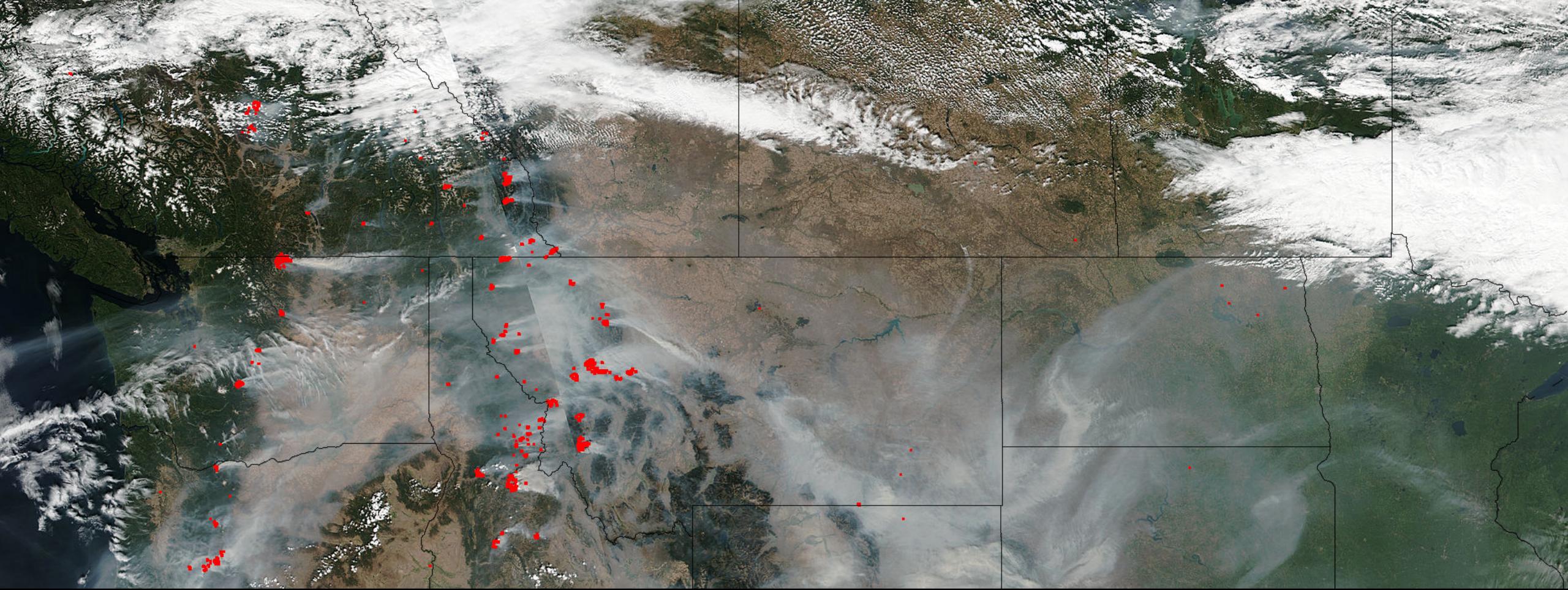
```
21
22 #This uses the file "fileList.txt", containing the list of files, in order to read the file
23 try:
24     fileList=open('fileList.txt','r')
25 except:
26     print('Did not find a text file containing file names (perhaps name does not match)')
27     sys.exit()
28
29 #loops through all files listed in the text file
30 for FILE_NAME in fileList:
31     FILE_NAME=FILE_NAME.strip()
32     user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
33     if(user_input == 'N' or user_input == 'n'):
34         continue
35     else:
36         if '3K' in FILE_NAME: #then this is a 3km MODIS file
37             print('This is a 3km MODIS file. Saving... ')
38             #saves all the SDS to be outputted to ASCII in a dictionary
39             dataFields=dict([(1,'Optical_Depth_Land_And_Ocean'),(2,'Image_Optical_Depth_La
40             # The name of the SDS to read
41             elif 'L2' in FILE_NAME: #Same as above but for 10km MODIS file
42                 print('This is a 10km MODIS file. Saving... ')
43                 dataFields=dict([(1,'Deep_Blue_Aerosol_Optical_Depth_550_Land'),(2,'AOD_550_Da
44             else:
45                 print('The file :',FILE_NAME, ' is not a valid MODIS file (or is named incorre
46                 continue
47             try:
48                 # open the hdf file for reading
49                 hdf=SD.SD(FILE_NAME)
50             except:
51                 print('Unable to open file: \n' + FILE_NAME + '\n Skipping...')
52                 continue
53
```



Applications

- This is a sample code to read and extract the MODIS Level 2 aerosol data
- The code can be modified to extract multiple SDSs into a single .csv file
- The code can be easily modified to extract data over a certain region
- The output file can be opened in excel, or any other data analysis tool





Create Air Quality Maps

Create an Air Quality Map

read_aod_and_calculate_pm25.py

- Purpose: read a MODIS aerosol level 2 data file in HDF format and create a PM2.5 air quality category map using the relationship between AOD and PM2.5
- The code works for both 10 km and 3 km products

Disclaimer: This is just a sample code. The default AOD-PM2.5 relationship used here is the assumed relationship over the USA.

The users of this code are responsible for checking the validity of this relationship and encouraged to use local relationships for visualizing AQ in different parts of the world.

The screenshot shows the Spyder Python IDE interface. The left pane displays the code for `read_aod_and_calculate_pm25.py`. The right pane shows the Python console output and the generated air quality map.

Code Snippet:

```
1 #!/usr/bin/python
2
3 # Modules: pm25_modis.py
4
5 # Disclaimer: The code is for demonstration purposes only. Users are responsible to check for
6 # Authors: Justin Roberts-Pierrel, 2015
7 # Organization: NASA ARSET
8 # Purposes: To extract AOD data from a MODIS HDF4 file (or series of files), calculate PM 2.5
9 # See the README associated with this module for more information.
10
11
12
13
14
15 #Import necessary modules
16 from pyhdf import SD
17 import numpy as np
18 import sys
19 from mpl_toolkits.basemap import Basemap
20 import matplotlib.pyplot as plt
21 from matplotlib.colors import LinearSegmentedColormap
22
23 #This uses the file "fileList.txt", containing the list of files, in order to read the file
24 try:
25     fileList=open('fileList.txt','r')
26 except:
27     print("Did not find a text file containing file names (perhaps name does not match!)")
28     sys.exit()
29
30 #Loop through all files listed in the text file
31 for FILE_NAME in fileList:
32     FILE_NAME=FILE_NAME.rstrip()
33     user_input=input("Would you like to process "+FILE_NAME+" (Y/N)?")
34     if user_input == 'N' or user_input == 'n':
35         continue
36     else:
37         if '3K' in FILE_NAME: #Then this is a 3km MODIS file
38             print("This is a 3km MODIS file. Here is some information: ")
39             SDS_NAME="Optical_Depth_Land_And_Ocean" #The name of the SDS to read
40         elif '12K' in FILE_NAME: #Same as above but for 12km MODIS file
41             print("This is a 12km MODIS file. Here is some information: ")
42             SDS_NAME="AOD_558_Dark_Target_Day_Combined"
43         else: #It is neither 3km nor 12km, then this will skip the rest of this loop
44             print("The file "+FILE_NAME+" is not a valid MODIS file OR is named incorrectly")
45             continue
46
47     #Open the HDF file for reading
48     try:
49         SD=SD(FILE_NAME)
50     except:
51         print("Unable to open file: "+FILE_NAME+" (Skipping...)")
```

Console Output:

```
Would you like to process MYD04_L2.A2017249.1925.006.2017250160408.hdf
(Y/N)Y
This is a 12km MODIS file. Here is some information:
(The valid range of values is: -0.1, " to ", 5.0, "(the average is: 0.178, (the standard deviation is: 0.23)
(The range of latitude in this file is: 27.187273, " to ", 48.299458, "degrees (The range of longitude in this file is: -111.39777, " to ", -88.255447, " degrees)

Would you like to enter a slope and intercept for PM 2.5 calculation?N

Would you like to create a map of this data? Please enter Y or N
Y
/MYD04_L2.A2017249.1925.006.2017250160408.hdf:1:UserWarning: The ishold() function was deprecated in version 2.4.
b = ax.ishold()
/MYD04_L2.A2017249.1925.006.2017250160408.hdf:1:UserWarning: axes.hold is deprecated.
See the API Changes document (http://matplotlib.org/api/api\_changes.html)
for more details.
ax.hold(b)
```

Map Output:

output



Editing the Code – Change the SDS

The user can change the AOD SDS to be used in PM_{2.5} calculation

```
-- 30 #loops through all files listed in the text file
31 for FILE_NAME in fileList:
32     FILE_NAME=FILE_NAME.strip()
33     user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
34     if(user_input == 'N' or user_input == 'n'):
35         continue
36     else:
37         if '3K' in FILE_NAME:#then this is a 3km MODIS file
38             print('This is a 3km MODIS file. Here is some information: ')
39             SDS_NAME='Optical_Depth_Land_And_Ocean' # The name of the sds to read
40         elif 'L2' in FILE_NAME: #Same as above but for 10km MODIS file
41             print('This is a 10km MODIS file. Here is some information: ')
42             SDS_NAME='AOD_550_Dark_Target_Deep_Blue_Combined'
43         else:#if it is neither 3km nor 10km, then this will skip the rest of this loop iteration
44             print('The file :',FILE_NAME, ' is not a valid MODIS file (Or is named incorrectly). \n')
45             continue
46         try:
47             # open the hdf file for reading
48             hdf=SD.SD(FILE_NAME)
49         except:
50             print('Unable to open file: \n' + FILE_NAME + '\n Skipping...')
51             continue
52
```



Editing the Code: Change the AOD-PM_{2.5} Relationship and AQI

The code uses

$$\text{PM2.5} = \text{Slope} * \text{AOD} + \text{Intercept}$$

as the linear regression equation
to calculate PM2.5 from AOD

The code uses the U.S. EPA
definition of air quality categories
based on PM2.5

AQI Calculator:

[https://airnow.gov/index.cfm?
action=airnow.calculator](https://airnow.gov/index.cfm?action=airnow.calculator)

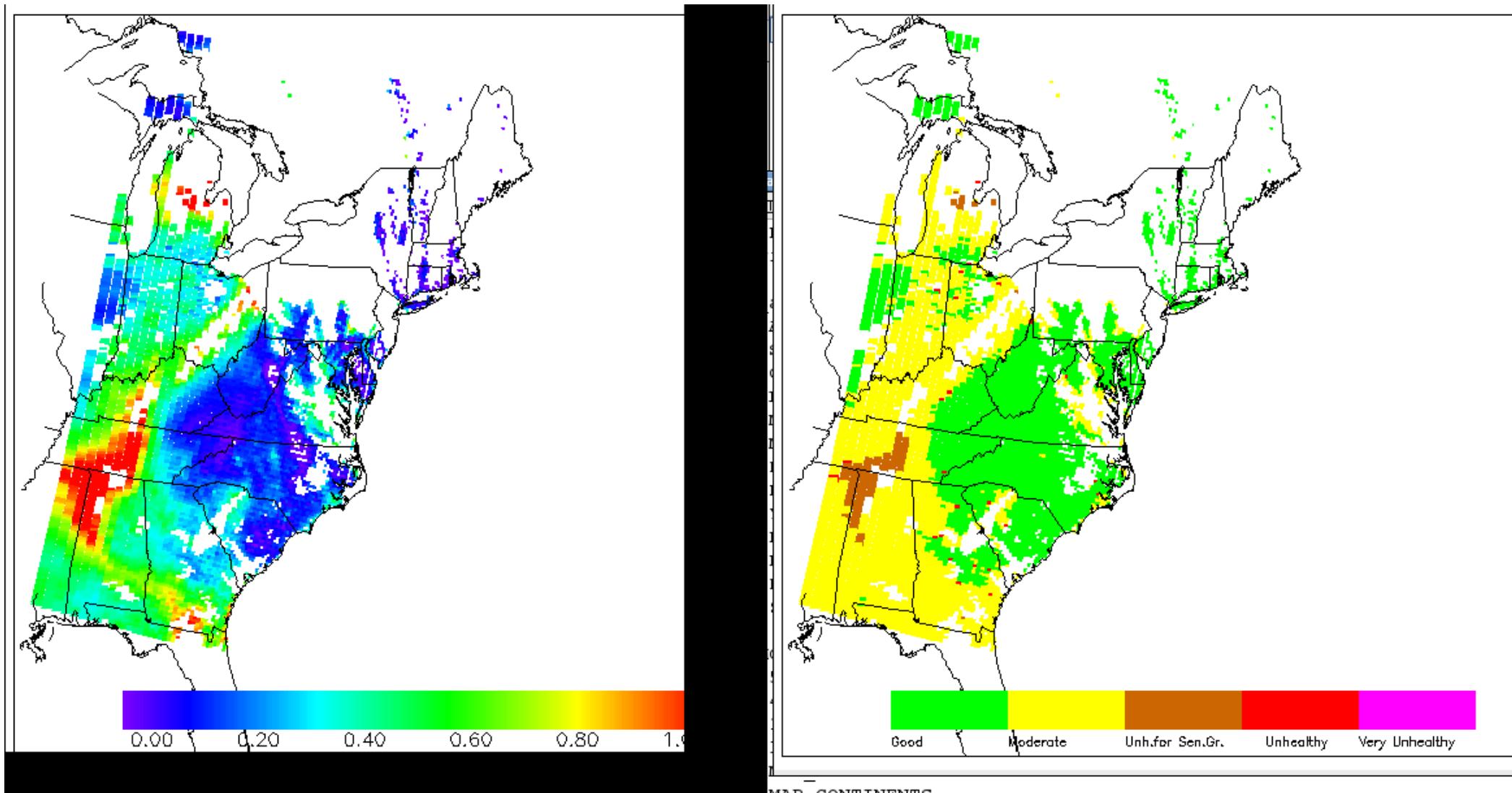
```
77
98 #asks user if they want to set PM2.5 calculation parameters
99 user_input=input('\nWould you like to enter a slope and intercept for PM 2.5 calculation?')
100 if user_input == 'Y' or user_input == 'y':
101     slope=input('Please enter a slope: ')
102     intercept=input('Please enter an intercept: ')
103 else:
104     #if not, choose the following:
105     slope=29.4
106     intercept=8.8
107     valid_data=data*scale_factor
108     pm25=float(slope)*valid_data+float(intercept)
109
110
111
112 #Asks user if they would like to see a map
113 is_map=input('\nWould you like to create a map of this data? Please enter Y or N \n')
114 #if user would like a map, view it
115 if is_map == 'Y' or is_map == 'y':
116     #turn fillvalues to NaN
117     data=pm25.astype(float)
118     data[np.logical_and(data>=0,data <= 12)]=0
119     data[np.logical_and(data>12,data <= 35.4)]=1
120     data[np.logical_and(data>35.4,data <= 55.4)]=2
121     data[np.logical_and(data>55.4,data <= 150.4)]=3
122     data[np.logical_and(data>150.4,data <= 250.4)]=4
123     data[data>250.4]=5
124     data[data < 0] = np.nan
125     #create the map
126     data = np.ma.masked_array(data, np.isnan(data))
127     m = Basemap(projection='cyl', resolution='l', llcrnrlat=min_lat, urcrnrlat = max_lat, llcrnrlon=min_lon,
128                 urcrnrlon=max_lon, lon_0=0)
129     m.drawcoastlines(linewidth=0.5)
130     m.drawparallels(np.arange(-90., 120., 5.), labels=[1, 0, 0, 0])
131     m.drawmeridians(np.arange(-180., 180., 5.), labels=[0, 0, 0, 1])
132     x, y = m(longitude, latitude)
133     my_cmap=LinearSegmentedColormap.from_list('mycmap', ['green','yellow','orange','red','purple','brown'],6)
134     m.pcolormesh(x, y, data,cmap=my_cmap)
135     plt.clim(0,6)
136     #create colorbar
137     cb = m.colorbar()
138     cb.set_label('AQI Category')
139     cb.set_ticks([.5, 1.5, 2.5, 3.5, 4.5, 5.5]) # force there to be only 7 ticks
140     cb.set_ticklabels(['Good', 'Moderate', 'Unhealthy for Sensitive Groups', 'Unhealthy', 'Very Unhealthy', 'Hazardous'])
```

Change the
default slope &
intercept

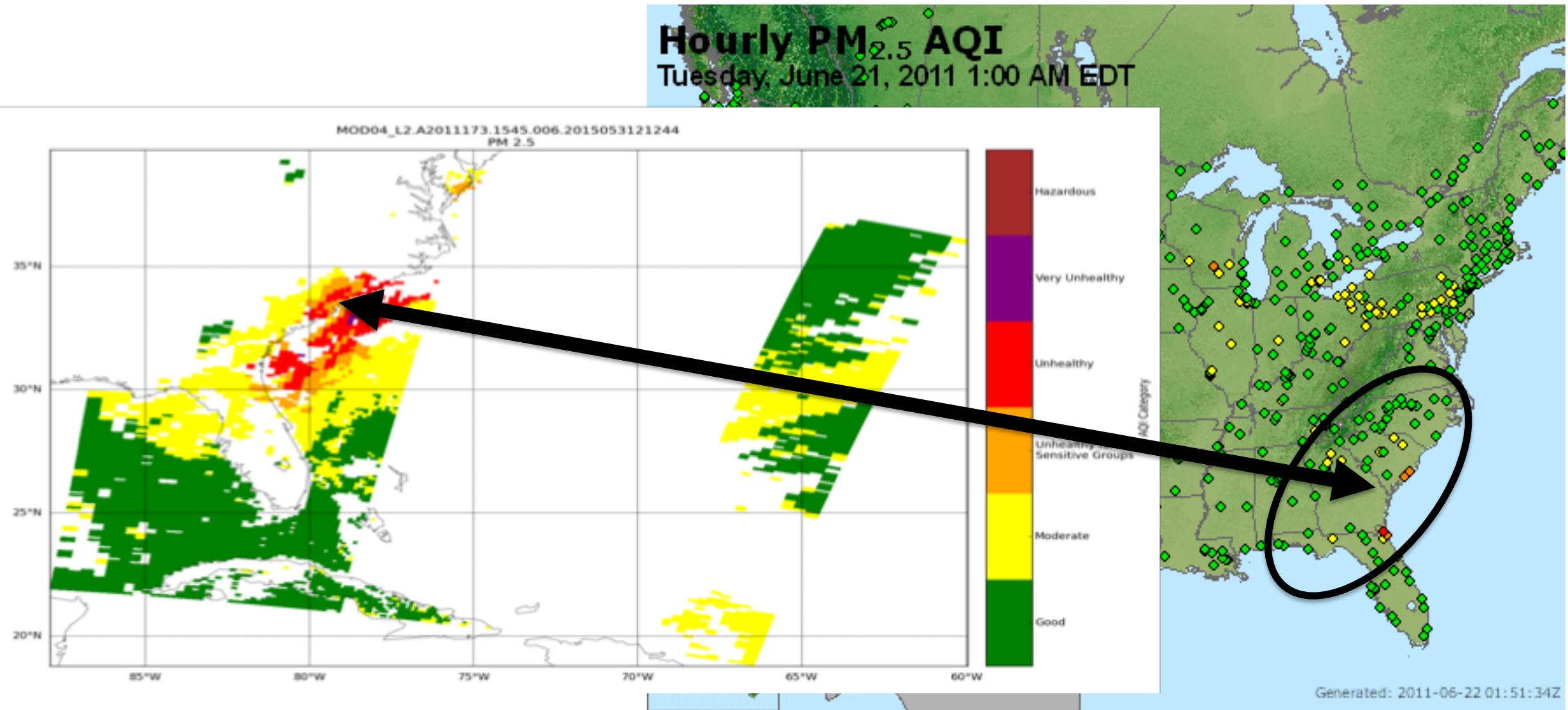
Change the air
quality categories



Application: Convert AOD into PM_{2.5} & Air Quality Maps



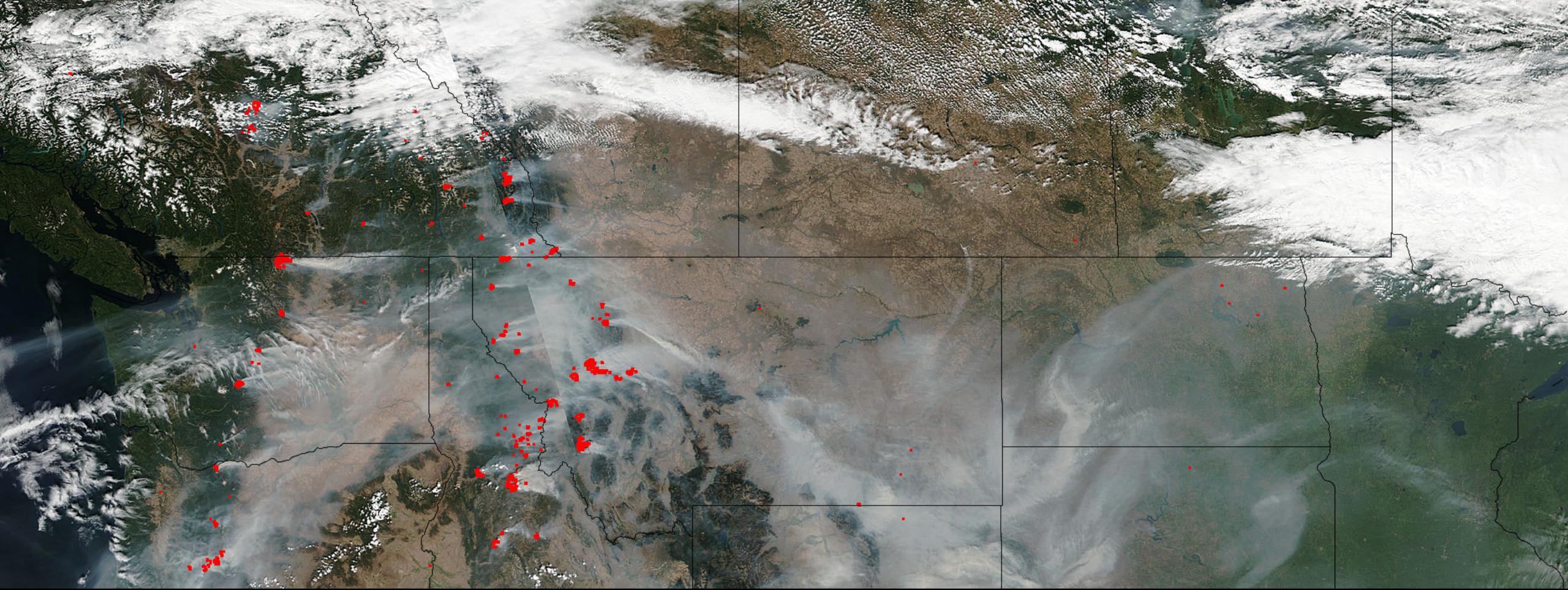
Application – Compare Satellite with Surface Maps



References

- Deep Blue Algorithm Website (<https://deepblue.gsfc.nasa.gov/data>)
- Dark Target Algorithm Website (<https://darktarget.asfc.nasa.gov/>)
- MODIS Data Download (<https://ladsweb.modaps.eosdis.nasa.gov/>)





Questions?