

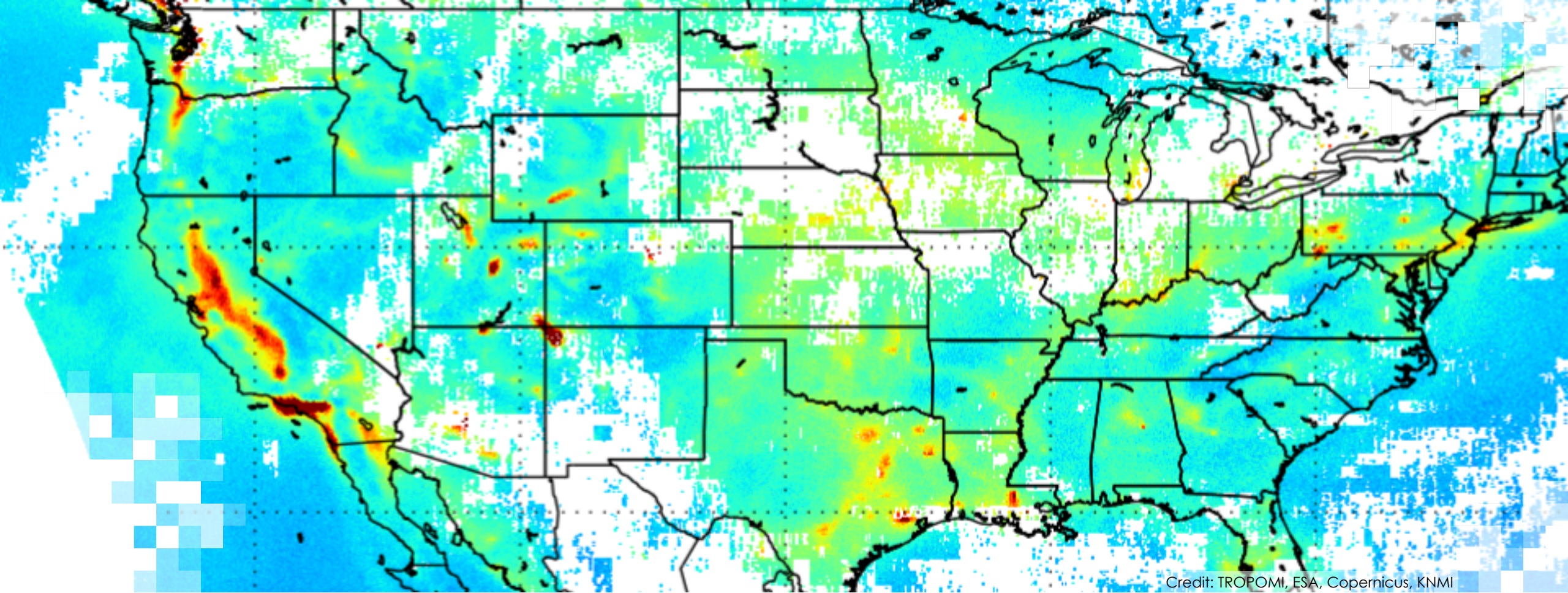
Credit: TROPOMI, ESA, Copernicus, KNMI

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

Pawan Gupta, and Melanie Follette-Cook

Application of Satellite Observations for Air Quality and Health Exposure, October 9-11, 2019

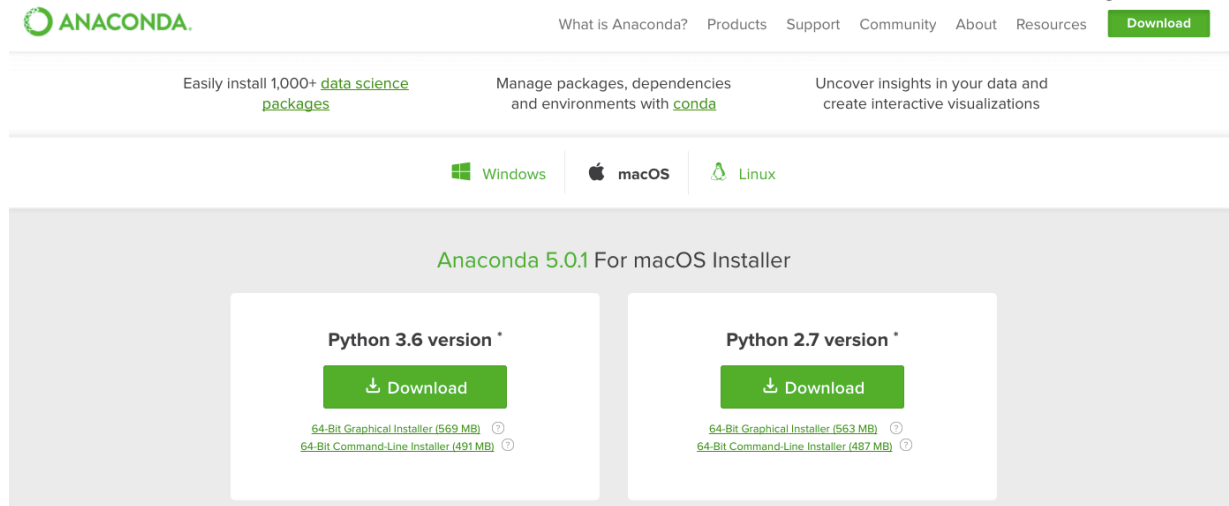




Computer and Python Requirements

Computer Requirements

- Install Python 3.7 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.gsfc.nasa.gov/airquality/python-scripts-aerosol-data-sets-merra-modis-and-omi>

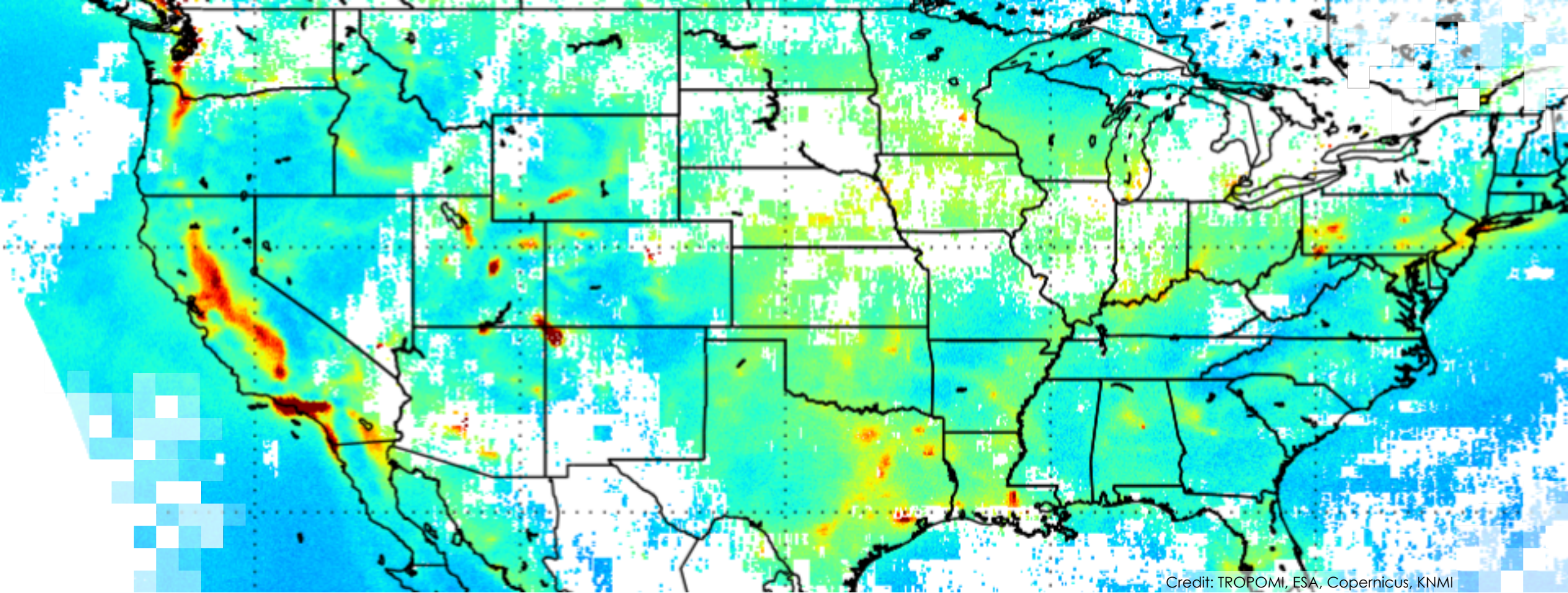


- Python package list:
 - pyhdf
 - numpy
 - sys
 - mpl_toolkits.
basemap
 - matplotlib
 - linearSegmented
Colormap
 - h5py
 - time
 - calendar

Python Test

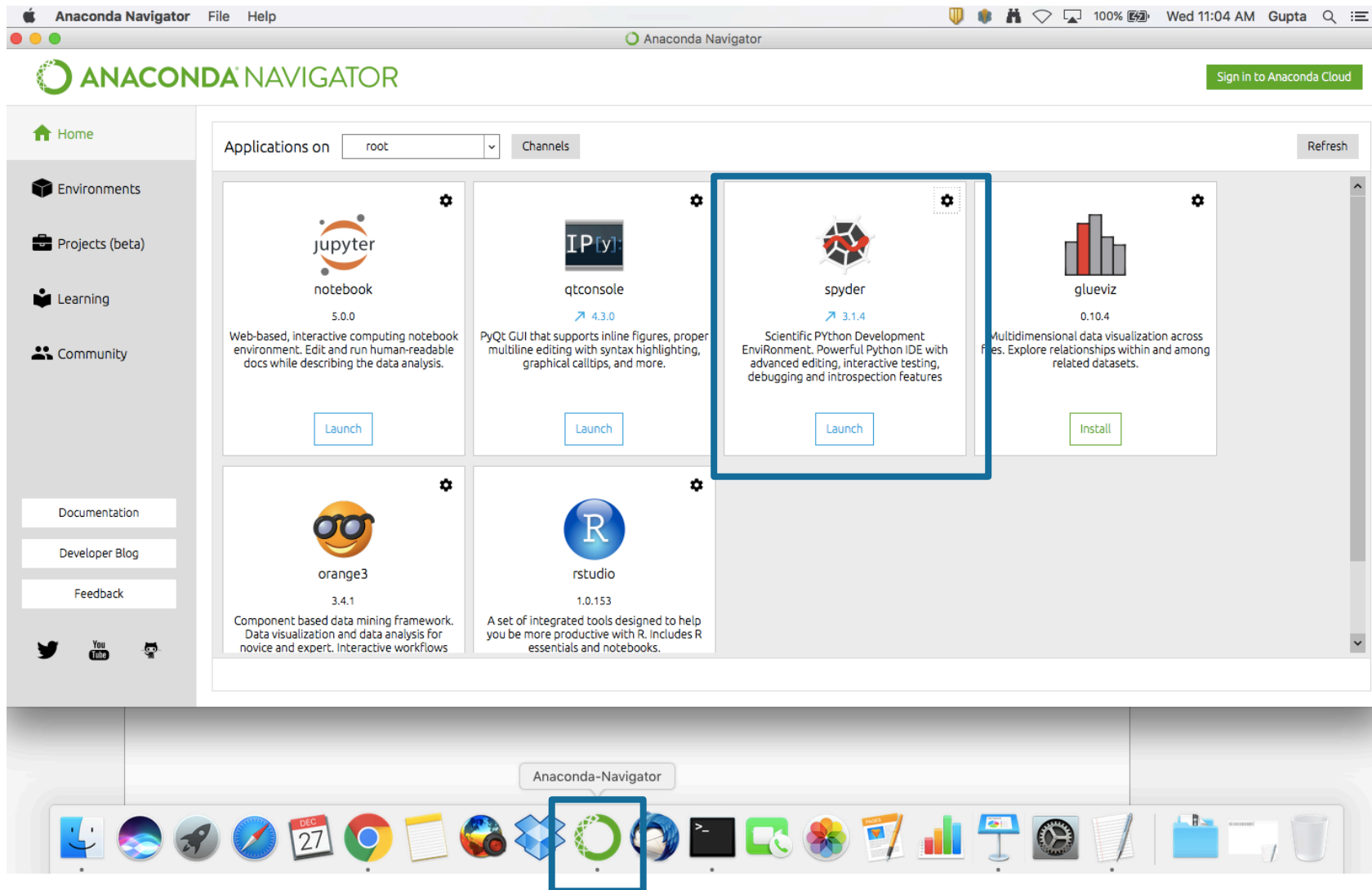
- Open the spyder editor inside Anaconda
- Open **test_python.py**
- Make sure the directory has the Python code and HDF file
- Open the **ipython** console in the spyder
- Run the code using the **green arrow** on the top
- Output should be an image as shown

The screenshot displays the Spyder Python IDE interface. The top menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, and Help. The main editor window shows a Python script named `test_python.py`. A green play button icon in the toolbar is circled in blue, with an arrow pointing to it. A blue box highlights the line `26 FILE_NAME='MYD04_L2.A2017249.2105.006.2017250160535.hdf'`, with an arrow pointing to it and the text "HDF file". The IPython console on the right shows the execution of the script, resulting in a map titled "MYD04_L2.A2017249.2105.006.2017250160535 Image Optical Depth Land And Ocean". The map shows a geographical area with a color scale for AOD (Aerosol Optical Depth) ranging from 0 to 5. An arrow points to the map with the text "output".



Getting Ready with Python

Anaconda & the Spyder Editor



Spyder View

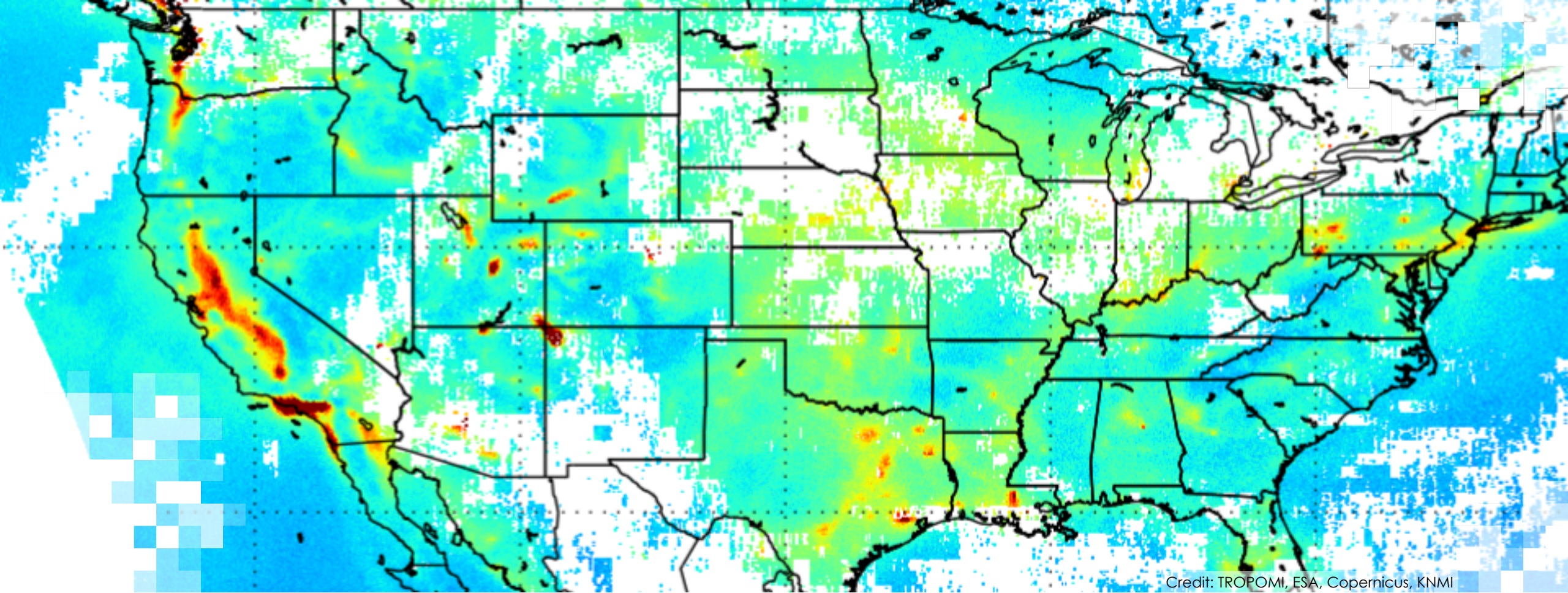
code area

ipython area

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

```
python File Edit Search Source Run Debug Consoles Projects Tools View Help
Spyder (Python 2.7)
/Users/gupta
read_mod_aerosol_and_list_sds.py
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-Pierel, 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\n(Y/N)')
29     if(user_input == 'N' or user_input == 'n'):
30         continue
31     else:
32         if '3K' in FILE_NAME: #then this is a 3km MODIS file
33             print('\nThis is a MODIS 3km file. Here is a list of SDS in your file:\n
34         elif 'L2' in FILE_NAME:
35             print('\nThis is a MODIS 10km file. Here is a list of SDS in your file:\n
36         else:
37             print('The file named :',FILE_NAME, ' is not a valid MODIS file (Or is i
38             continue
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 hdf4 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
```

Usage

Here you can get help of any object by pressing **Cmd+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in **Preferences > Help**.

New to Spyder? Read our [tutorial](#)

Variable explorer | File explorer | Help | Profiler

IPython console

Console 1/A

```
Python 2.7.13 [Anaconda 4.4.0 (x86_64)] (default, Dec 20 2016, 23:05:08)
Type "copyright", "credits" or "license" for more information.

IPython 5.3.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

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
MYD04_L2.A2017249.1925.006.2017250160408.hdf
(Y/N)"Y"
```

Python console | History log | IPython console

Permissions: RW End-of-lines: CR Encoding: ASCII Line: 1 Column: 1 Memory: 73 %



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 displays the Spyder Python IDE interface. The top toolbar features a green play button, which is circled in blue and has a mouse cursor pointing to it. The main editor window shows a Python script named `read_mod_aerosol_and_list_sds.py`. The script includes a disclaimer, author information (Justin Roberts-Pierel, 2015), and code that reads a text file (`fileList.txt`) to process MODIS files. The code prompts the user to confirm processing each file. The IPython console at the bottom shows the execution output, including the prompt `(Y/N) "Y"` and a list of 23 SDS names for a MODIS 10km file. A blue box highlights the console output area, with the word "output" written in large black text to its right. The console also shows a "Usage" help window and a "New to Spyder? Read our tutorial" message.

```
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-Pierel, 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\n(Y/N)')
29     if(user_input == 'N' or user_input == 'n'):
30         continue
31     else:
32         if '3K' in FILE_NAME: #then this is a 3km MODIS file
33             print('\nThis is a MODIS 3km file. Here is a list of SDS in your file:
34         elif 'L2' in FILE_NAME:
35             print('\nThis is a MODIS 10km file. Here is a list of SDS in your file
36         else:
37             print('The file named :,FILE_NAME, ' is not a valid MODIS file (Or is
38             continue
39     try:
40         # open the hdf file for reading
41         hdf=SD.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 hdf4 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
```

Usage

Here you can get help of any object by pressing **Cmd+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in **Preferences > Help**.

New to Spyder? Read our [tutorial](#)

Variable explorer File explorer Help Profiler

IPython console

Console 1/A

(Y/N) "Y"

This is a MODIS 10km 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_2_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_TOA_Reflectance_Land

Python console History log IPython console

Permissions: RW End-of-lines: CR Encoding: ASCII Line: 1 Column: 1 Memory: 37 %

Editing the Code

```
Module: read_mod_aerosol_and_lst_sds.py
=====
Disclaimer: The code is for demonstration purposes only. Users are responsible to check for accuracy and revise to fit their objectives.
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 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()

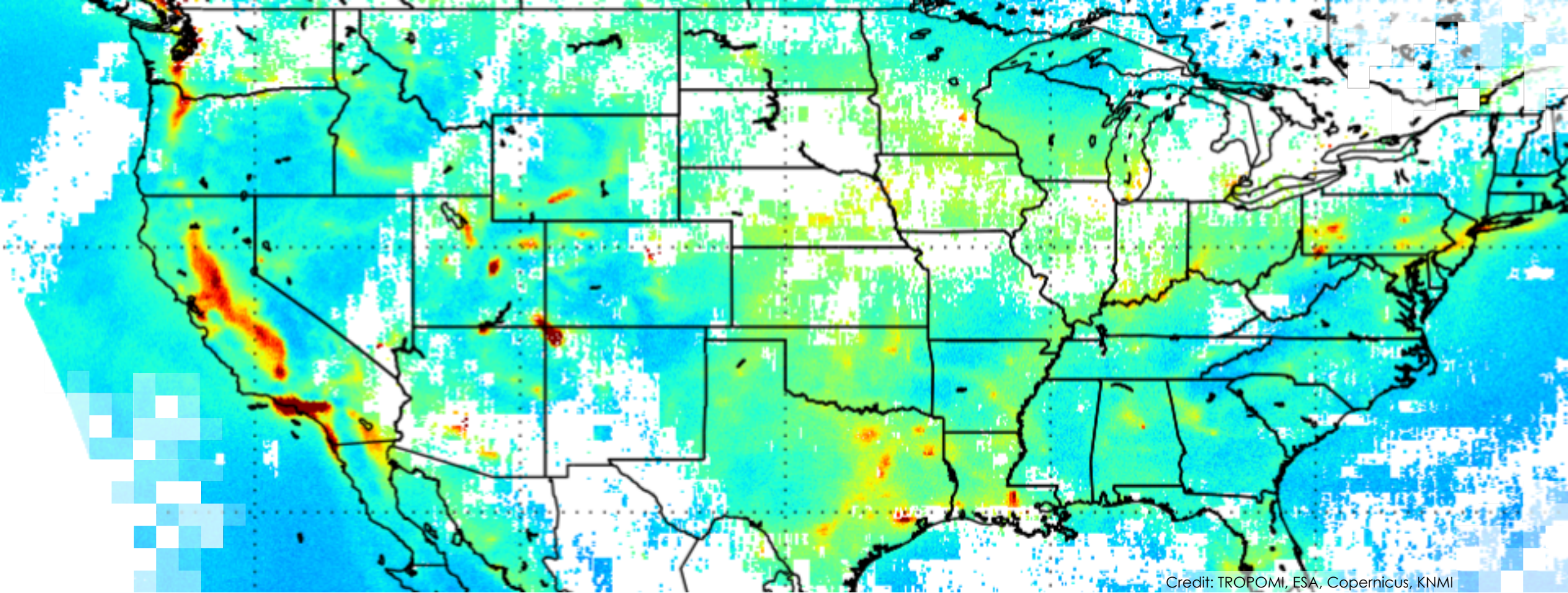
#loops through all files listed in the text file
for FILE_NAME in fileList:
    FILE_NAME=FILE_NAME.strip()
    user_input=input('Would you like to process\n' + FILE_NAME + '\n\n(Y/N)')
    if(user_input == 'N' or user_input == 'n'):
        continue
    else:
        if '3K' in FILE_NAME: #then this 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

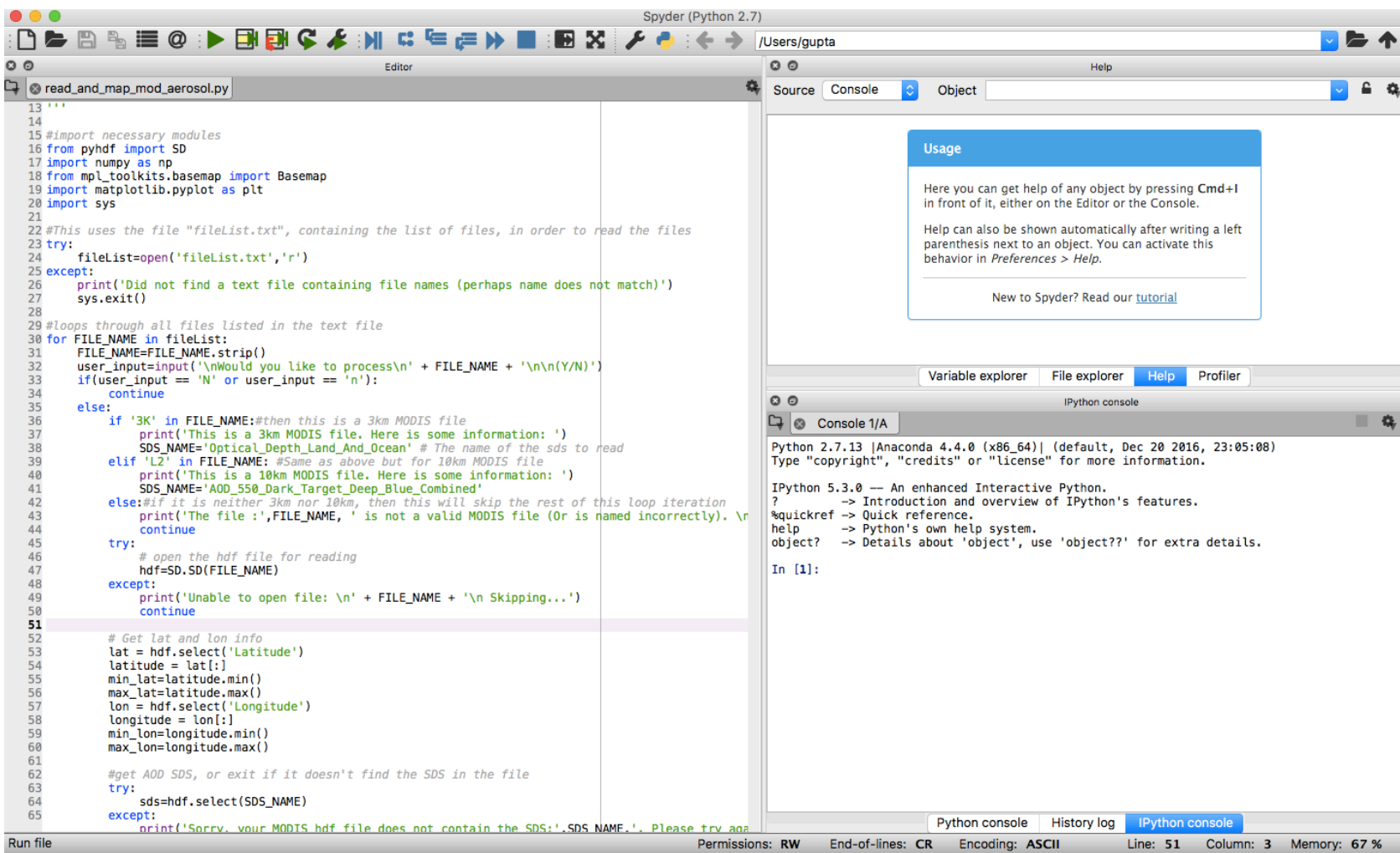
- MODIS Level 2 aerosol data are provided in HDF files
- Each HDF file contains several geophysical parameters
- Special codes and tools are required to open HDF files
- This code helps users see the name of the available SDSs inside an HDF file for further analysis



Map Aerosol Optical Depth

Plot and save a map of MODIS AOD

read_and_map_mod_aerosol.py



The image shows the Spyder Python IDE interface. The main editor window displays the script `read_and_map_mod_aerosol.py`. The script includes comments and code for reading MODIS files, processing AOD data, and calculating latitude and longitude. A help window is open over the console, providing usage instructions for the IDE.

```
13 '''
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 #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. Here is some information: ')
38             SDS_NAME='Optical_Depth_Land_And_Ocean' # The name of the sds to read
39         elif 'L2' 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).\n')
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 aga
```



Running and Output

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

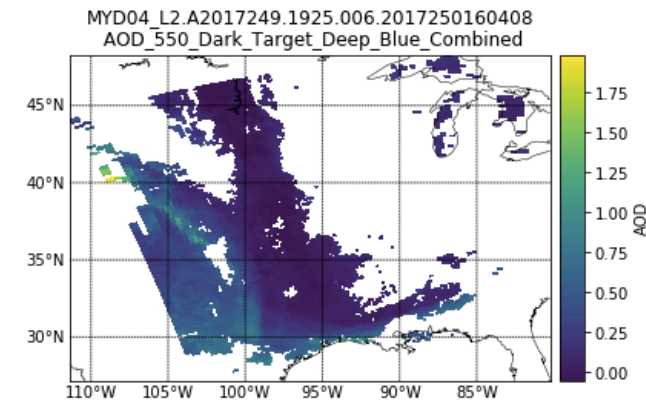
```
Would you like to process  
MYD04_L2.A2017249.1925.006.2017250160408.hdf
```

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

AOD statistics

```
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 ishield function was deprecated in version 2.0.  
    b = ax.ishield()  
/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
```

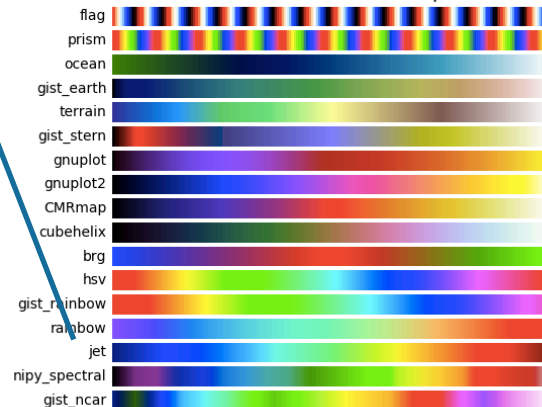

Editing the Code

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., 181., 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')
```

(png, pdf)

Miscellaneous colormaps



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

Change the SDS

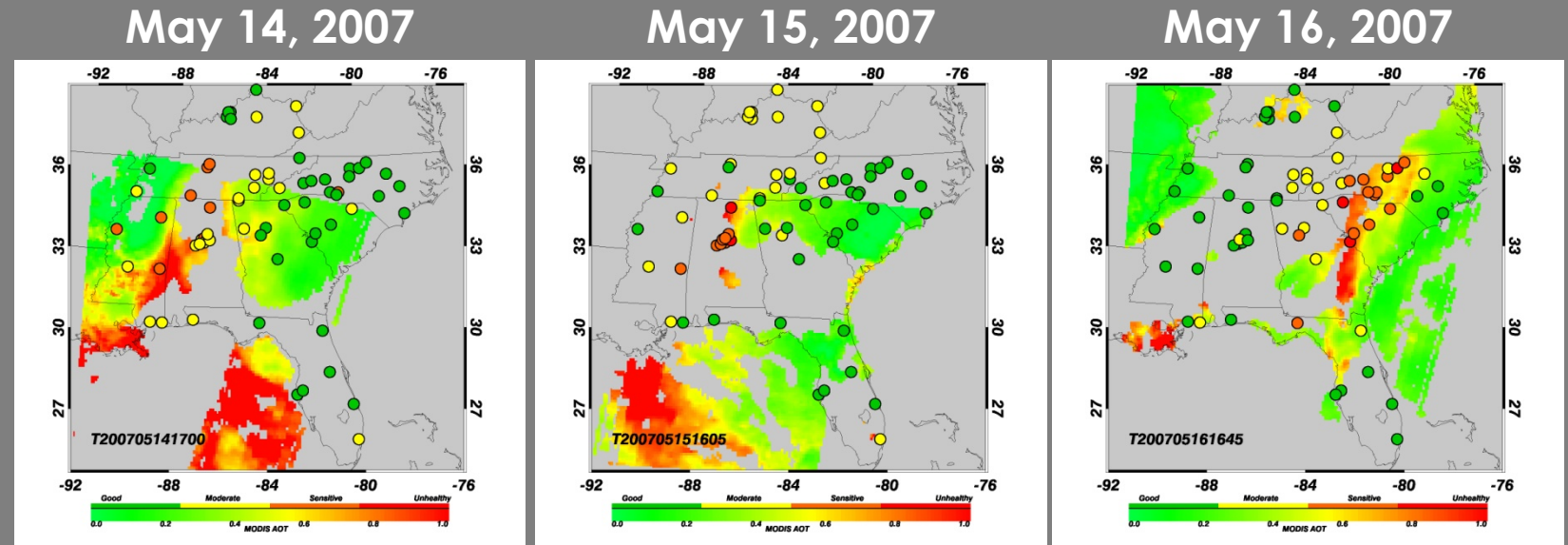
```
#loops through all files listed in the text file
for FILE_NAME in fileList:
    FILE_NAME=FILE_NAME.strip()
    user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
    if(user_input == 'N' or user_input == 'n'):
        continue
    else:
        if '3K' in FILE_NAME:#then 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_550_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
```

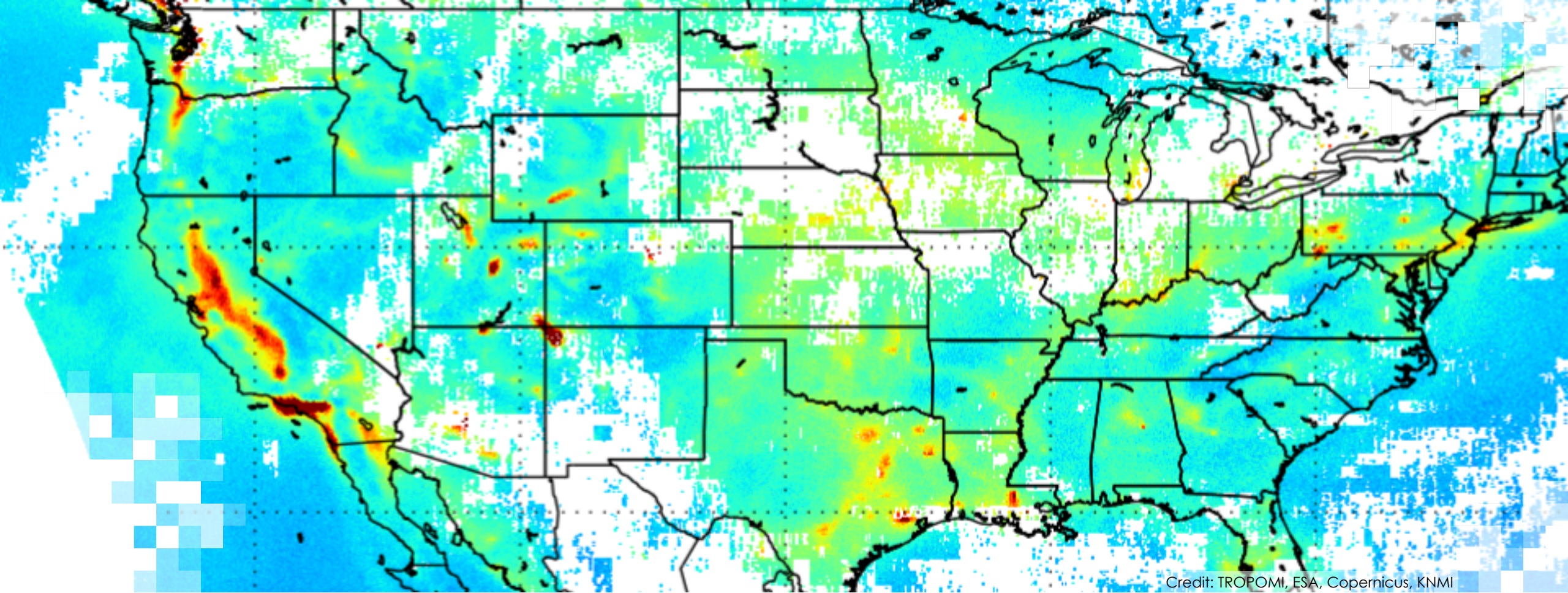
Applications

- 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

```
1 #!/usr/bin/python
2 '''
3 Module: read_mod_aerosol_at_a_location.py
4
5 Disclaimer: The code is for demonstration purposes only. Users are responsible to check for accuracy and
6
7 Author: Justin Roberts-Pierel, 2015
8 Organization: NASA ARSET
9 Purpose: To view info about a variety of SDS from a MODIS HDF4 file (or series of files) both generally and
10
11 See the README associated with this module for more information.
12 '''
13
14
15 #import necessary modules
16 from pyhdf import SD
17 import numpy as np
18
19 #This uses the file "fileList.txt", containing the list of files, in order to read the files
20 try:
21     fileList=open('fileList.txt','r')
22 except:
23     print('Did not find a text file containing file names (perhaps name does not match)')
24     sys.exit()
25
26 #loops through all files listed in the text file
27 for FILE_NAME in fileList:
28     FILE_NAME=FILE_NAME.strip()
29     user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
30     if(user_input == 'N' or user_input == 'n'):
31         continue
32     else:
33         if '3K' in FILE_NAME: #then this is a 3km MODIS file
34             userInput=int(input('Which SDS would you like to view? (Type the number and press enter) \n(
35             while userInput not in {1,2,3,4}:#repeats the question if the user does not choose one of the
36                 print('Please try again.')
37             userInput=int(input('Which SDS would you like to view? (Type the number and press enter)
38             #Uses a Python dictionary to choose the SDS indicated by the user
39             dataFields=dict([(1,'Optical_Depth_Land_And_Ocean'),(2,'Land_Ocean_Quality_Flag'),(3,'Image_
40             elif 'L2' in FILE_NAME:#Same as above but for 10km MODIS file
41             userInput=int(input('Which SDS would you like to view? (Type the number and press enter) \n(
42             while userInput not in {1,2,3}:
43                 print('Please try again.')
44             userInput=int(input('Which SDS would you like to view? (Type the number and press enter)
45             dataFields=dict([(1,'Deep_Blue_Aerosol_Optical_Depth_550_Land'),(2,'AOD_550_Dark_Target_Deep_
46             SDS_NAME=dataFields[int(userInput)] # The name of the sds to read
47             try:
48                 # open the hdf file for reading
49                 hdf=SD(FILE_NAME)
50             except:
51                 print('Unable to open file: \n' + FILE_NAME + '\n Skipping...')
52                 continue
53
```

degrees')
Please enter the latitude you would like to analyze (Deg. N): 40.0
Please enter the longitude you would like to analyze (Deg. E): -90.5
(\n\nThe nearest pixel to your entered location is at: \nLatitude:', 40.030121, '\nLongitude:', -90.550514)
(\n\nThe value of ', 'AOD_550_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
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
(\n\nThe range of latitude in this file is: ', 44.253548, ' to ', 66.211197, 'degrees
\n\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
(\n\nThe nearest pixel to your entered location is at: \nLatitude:', 49.482555, '\nLongitude:', -100.51669)
(\n\nThe value of ', 'Deep_Blue_Aerosol_Optical_Depth_550_Land', 'at this pixel is ', 0.171)
(\n\nThere', 'are', 9, 'valid', 'pixels', 'in a 3x3 grid centered at your entered location.)
(\n\nThe average value in this grid is: ', 0.197, '\n\nThe median value in this grid is: ', 0.171, '\n\nThe standard deviation in this grid is: ', 0.075)
(\n\nThere', 'are', 24, ' valid', 'pixels', ' in a 5x5 grid centered at your entered location. \n\n)
(\n\nThe average value in this grid is: ', 0.204, '\n\nThe median value in this grid is: ', 0.201, '\n\nThe standard deviation in this grid is: ', 0.077)

Would you like to process
MYD04_L2.A2017249.2105.006.2017250160535.hdf
(Y/N)

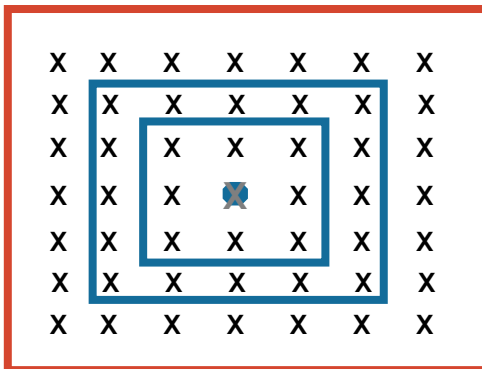
Running and Output

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

Select SDS

Lat & Lon of station

Outputs



```
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
('\n\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)
('\n\nThere', 'are', 9, 'valid', 'pixels', 'in a 3x3 grid centered at your entered
location.')
('\n\nThe average value in this grid is: ', 0.197, ' \n\nThe median value in this grid
is: ', 0.171, '\n\nThe standard deviation in this grid is: ', 0.075)
('\n\nThere', 'are', 24, ' valid', 'pixels', ' in a 5x5 grid centered at your entered
location. \n')
('The average value in this grid is: ', 0.204, ' \n\nThe median value in this grid
is: ', 0.201, '\n\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

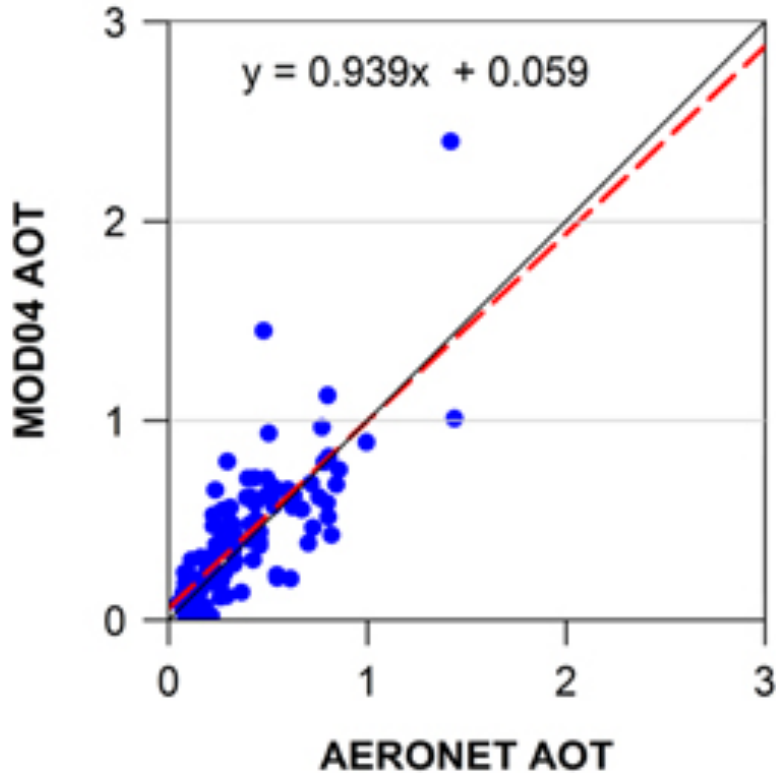
```
s through all files listed in the text file
ILE_NAME in fileList:
FILE_NAME=FILE_NAME.strip()
user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
if(user_input == 'N' or user_input == 'n'):
    continue
else:
    if '3K' in FILE_NAME: #then this is a 3km MODIS file
        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_Dep
        dataFields=dict([(1,'Deep_Blue_Aerosol_Optical_Depth_550_Land'),(2,'AOD_550_Dark_Target_Deep_Blue_Combined'),(3,'AOD_550_Dark_Tai
    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

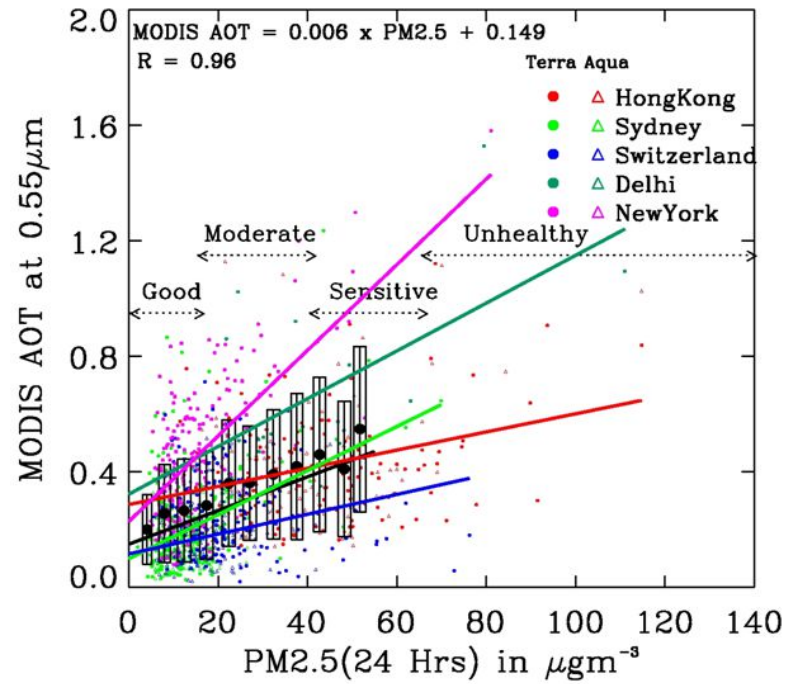
```
#calculates mean, median, stdev in a 3x3 grid around nearest point to entered location
if x < 1:
    x+=1
if x > data.shape[0]-2:
    x-=2
if y < 1:
    y+=1
if y > data.shape[1]-2:
    y-=2
three_by_three=data[x-1:x+2,y-1:y+2]
three_by_three=three_by_three.astype(float)
three_by_three[three_by_three==float(fillvalue)]=np.nan
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

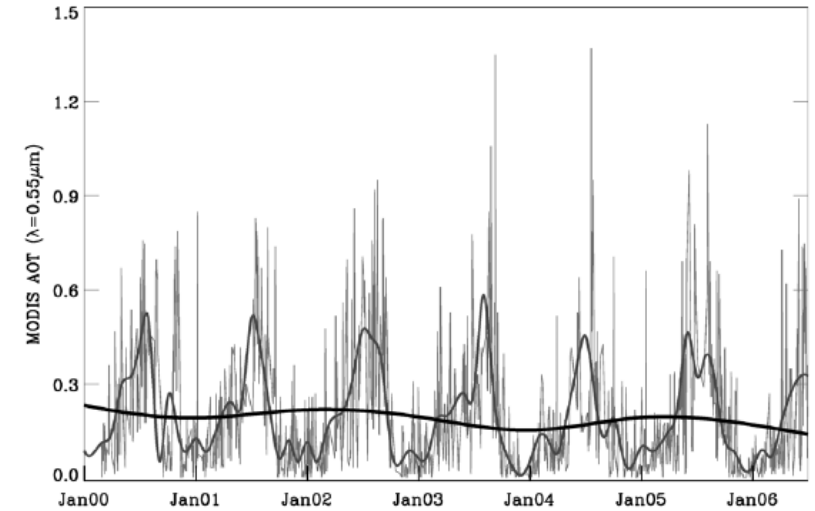
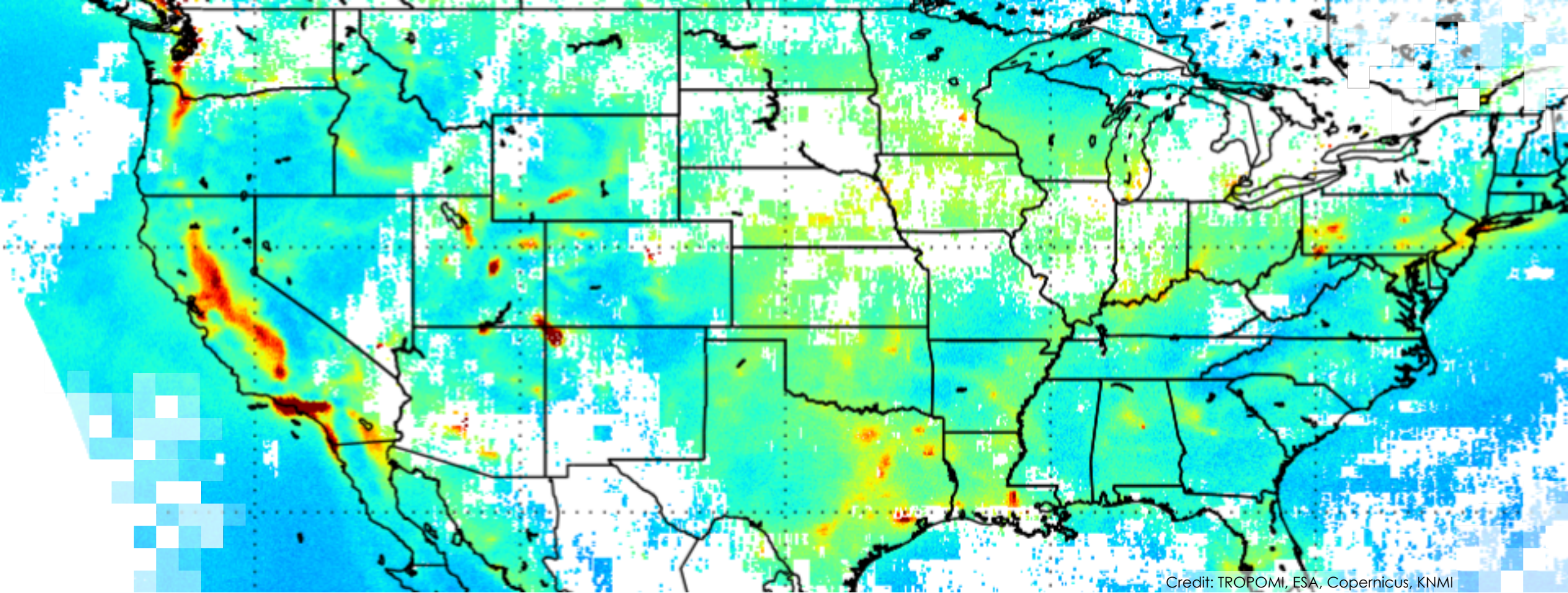


Image Sources: Gupta et al., Gupta et al., 2006, Gupta et al., 2007

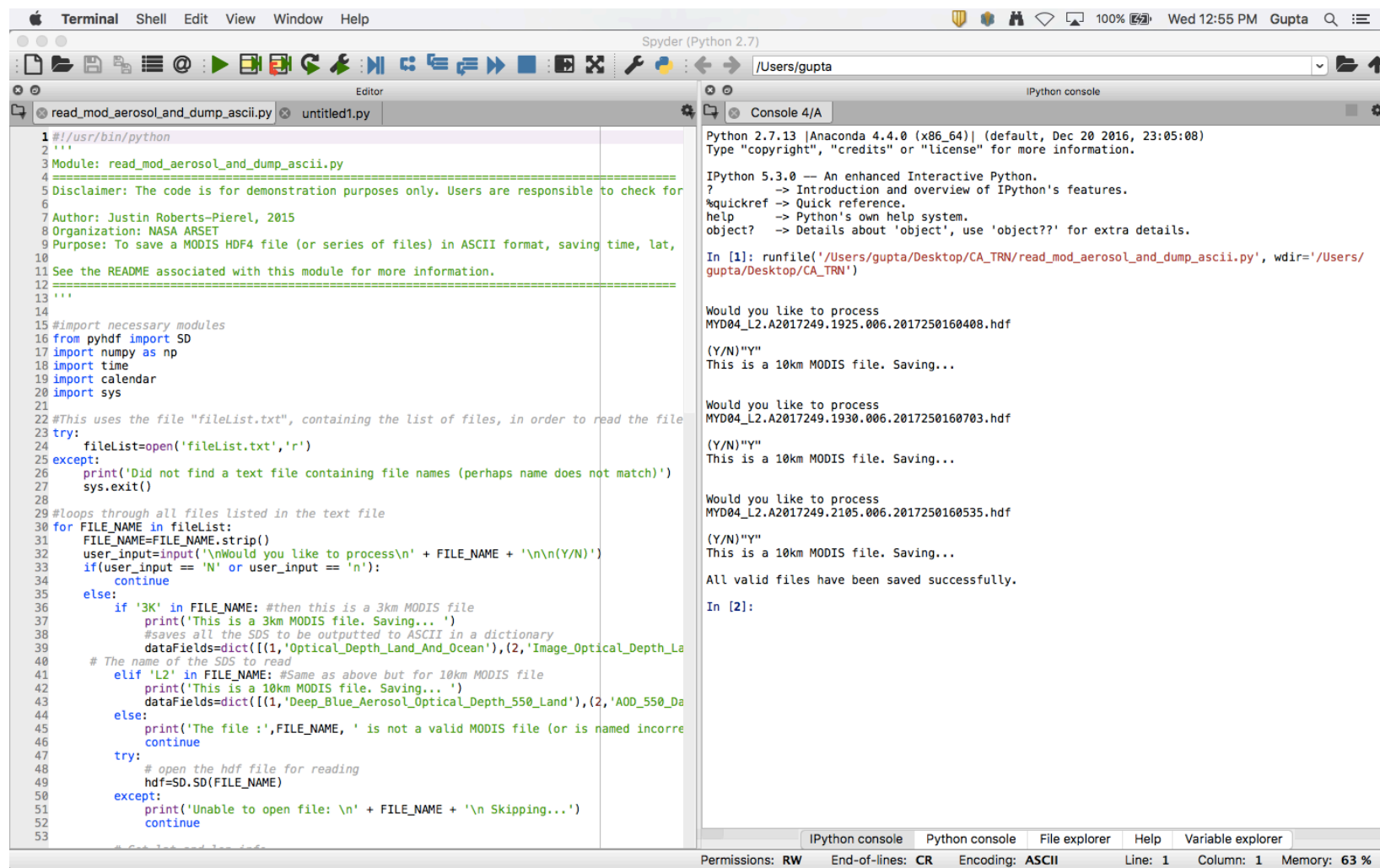


Output HDF Variables to CSV

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



```
Terminal Shell Edit View Window Help
Spyder (Python 2.7)
/Users/gupta
Editor
read_mod_aerosol_and_dump_ascii.py untitled1.py
1 #!/usr/bin/python
2 '''
3 Module: read_mod_aerosol_and_dump_ascii.py
4
5 Disclaimer: The code is for demonstration purposes only. Users are responsible to check for
6
7 Author: Justin Roberts-Pierel, 2015
8 Organization: NASA ARSET
9 Purpose: To save a MODIS HDF4 file (or series of files) in ASCII format, saving time, lat,
10
11 See the README associated with this module for more information.
12
13 '''
14
15 #import necessary modules
16 from pyhdf import SD
17 import numpy as np
18 import time
19 import calendar
20 import sys
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(FILE_NAME)
50             except:
51                 print('Unable to open file: \n' + FILE_NAME + '\n Skipping...')
52                 continue
53
```

```
Python 2.7.13 [Anaconda 4.4.0 (x86_64)] (default, Dec 20 2016, 23:05:08)
Type "copyright", "credits" or "license" for more information.

IPython 5.3.0 -- An enhanced Interactive Python.
?                -> Introduction and overview of IPython's features.
%quickref        -> Quick reference.
help             -> Python's own help system.
object?         -> Details about 'object', use 'object??' for extra details.

In [1]: runfile('/Users/gupta/Desktop/CA_TRN/read_mod_aerosol_and_dump_ascii.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. Saving...

Would you like to process
MYD04_L2.A2017249.1930.006.2017250160703.hdf
(Y/N)"Y"
This is a 10km MODIS file. Saving...

Would you like to process
MYD04_L2.A2017249.2105.006.2017250160535.hdf
(Y/N)"Y"
This is a 10km MODIS file. Saving...

All valid files have been saved successfully.

In [2]:

IPython console Python console File explorer Help Variable explorer
Permissions: RW End-of-lines: CR Encoding: ASCII Line: 1 Column: 1 Memory: 63 %
```


Output

```
MYD04_L2.A2017249.1925.006.2017250160408.txt
Year,Month,Day,Hour,Minute,Second,Longitude,Deep_Blue_Aerosol_Optical_Depth_550_Land,A00_550_Dark_Target_Deep_Blue_Combined,A00_550_Dark_Target_Deep_Blue_Combined_QA_Flag
2017.0,9.0,6.0,19.0,25.0,9.0,30.4542312622,-80.2554473877,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,30.4285984039,-80.7235641479,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,30.4032402039,-81.1592407227,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,30.3782196045,-81.5666427612,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,30.3535690300,-81.9493026733,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,30.329334259,-82.3096008301,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,30.3055496216,-82.6497421265,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,30.2822036743,-82.9720306396,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,30.2592868805,-83.2782287598,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,30.2367897034,-83.5699691772,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,30.2147369385,-83.8481216431,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,30.1930789948,-84.1143035889,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,30.1718177795,-84.3694152832,0.322000015294,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,30.1509361267,-84.6143875122,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,30.1304397583,-84.8497695923,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,30.1103076935,-85.0764007568,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,30.0905189514,-85.2949752808,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,30.0710792542,-85.5059127808,0.496000023559,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,30.0519447327,-85.7100372314,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,30.0331192017,-85.9076538086,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,30.0145874023,-86.0991973877,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.9963378906,-86.2850646973,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.9783554077,-86.4656219482,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.9606361389,-86.6411895752,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.9431610107,-86.8120880127,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.9259204865,-86.9785919189,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.9089050293,-87.1409683228,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.8921031952,-87.2994613647,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.87550354,-87.4542999268,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.8590984344,-87.6056747437,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.8428764343,-87.7537918091,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.8268318176,-87.8988342285,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.8109512329,-88.040977478,-9999.0,0.483000022941,1.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.7952270508,-88.1803741455,-9999.0,0.537000025506,3.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.7796516418,-88.3171691895,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.7642173767,-88.4515228271,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.7489128113,-88.5835418701,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.7337341309,-88.7133789062,-9999.0,0.523000024841,1.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.7186717987,-88.8411331177,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.7037162781,-88.9669265747,-9999.0,0.400000018999,1.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.688867569,-89.0908660889,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.6741104126,-89.2130584717,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.6594429016,-89.3335876465,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.6448535919,-89.452545166,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.6303424835,-89.5700378418,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.6158943176,-89.6861343384,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.6015090942,-89.8009262085,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.587179184,-89.9144897461,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.5728931427,-90.0269012451,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.5586509705,-90.1382369995,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.5444412231,-90.248550415,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.5302619934,-90.3579177856,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.5161094666,-90.4663772583,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.5019721985,-90.5740280151,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.4878482819,-90.6808853149,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.4737319946,-90.7870483398,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.4596118927,-90.8925476074,-9999.0,-9999.0,0.0
2017.0,9.0,6.0,19.0,25.0,9.0,29.4454841614,-90.9974596116,-9999.0,-9999.0,0.0
```

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



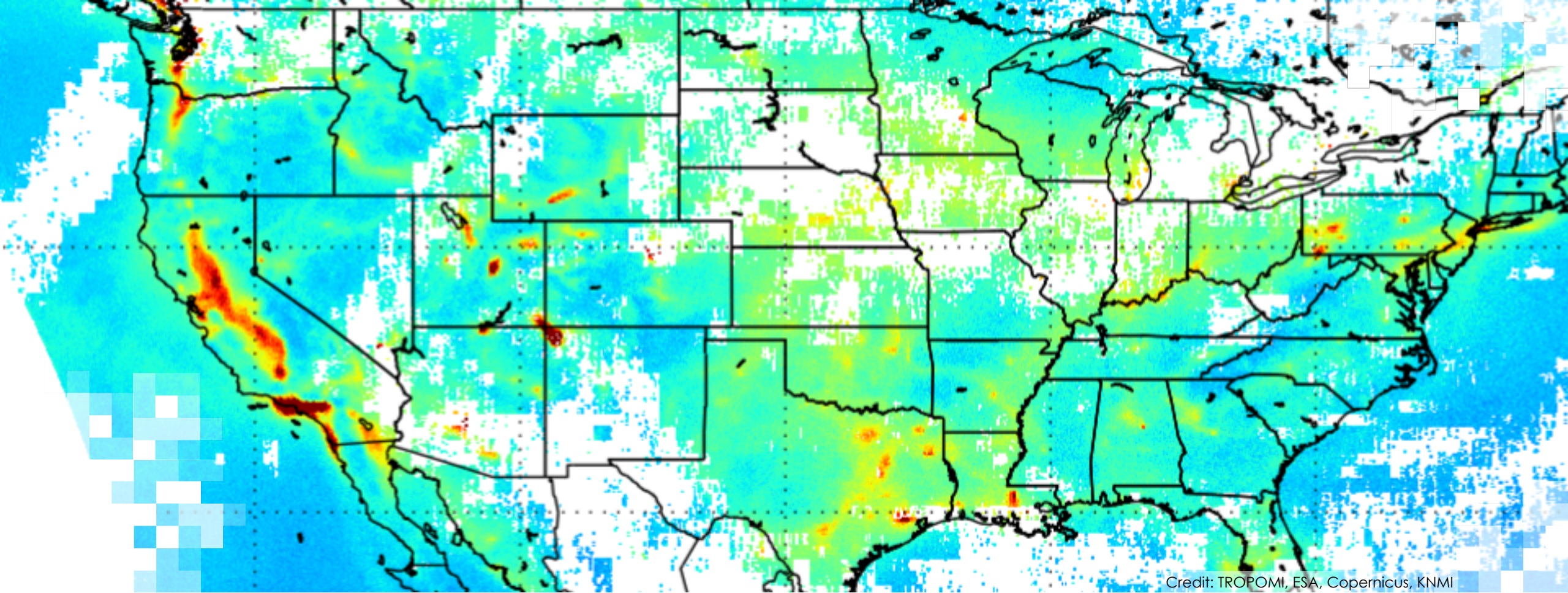
Editing the Code

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 PM_{2.5} air quality category map using the relationship between AOD and PM_{2.5}
- The code works for both 10 km and 3 km products

Disclaimer: This is just a sample code. The default AOD-PM_{2.5} 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 displays the Spyder Python IDE interface. The editor window shows the code for `read_aod_and_calculate_pm25.py`. The code includes a disclaimer, imports necessary modules (SD, numpy, sys, Basemap, plt, LinearSegmentedColormap), and a main loop that processes MODIS files. It prompts the user to process a specific file and to enter a slope and intercept for the PM_{2.5} calculation. The console window shows the execution output, including the file path `MYD04_L2.A2017249.1925.006.2017250160408.hdf` and the resulting map. The map shows the United States with an AQI legend ranging from Good (green) to Hazardous (red). The console also shows a warning about the deprecated `ax.hold(b)` function.

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

$PM_{2.5} = \text{Slope} * \text{AOD} + \text{Intercept}$
as the linear regression equation
to calculate PM_{2.5} from AOD

The code uses the U.S. EPA
definition of air quality categories
based on PM_{2.5}

AQI Calculator:

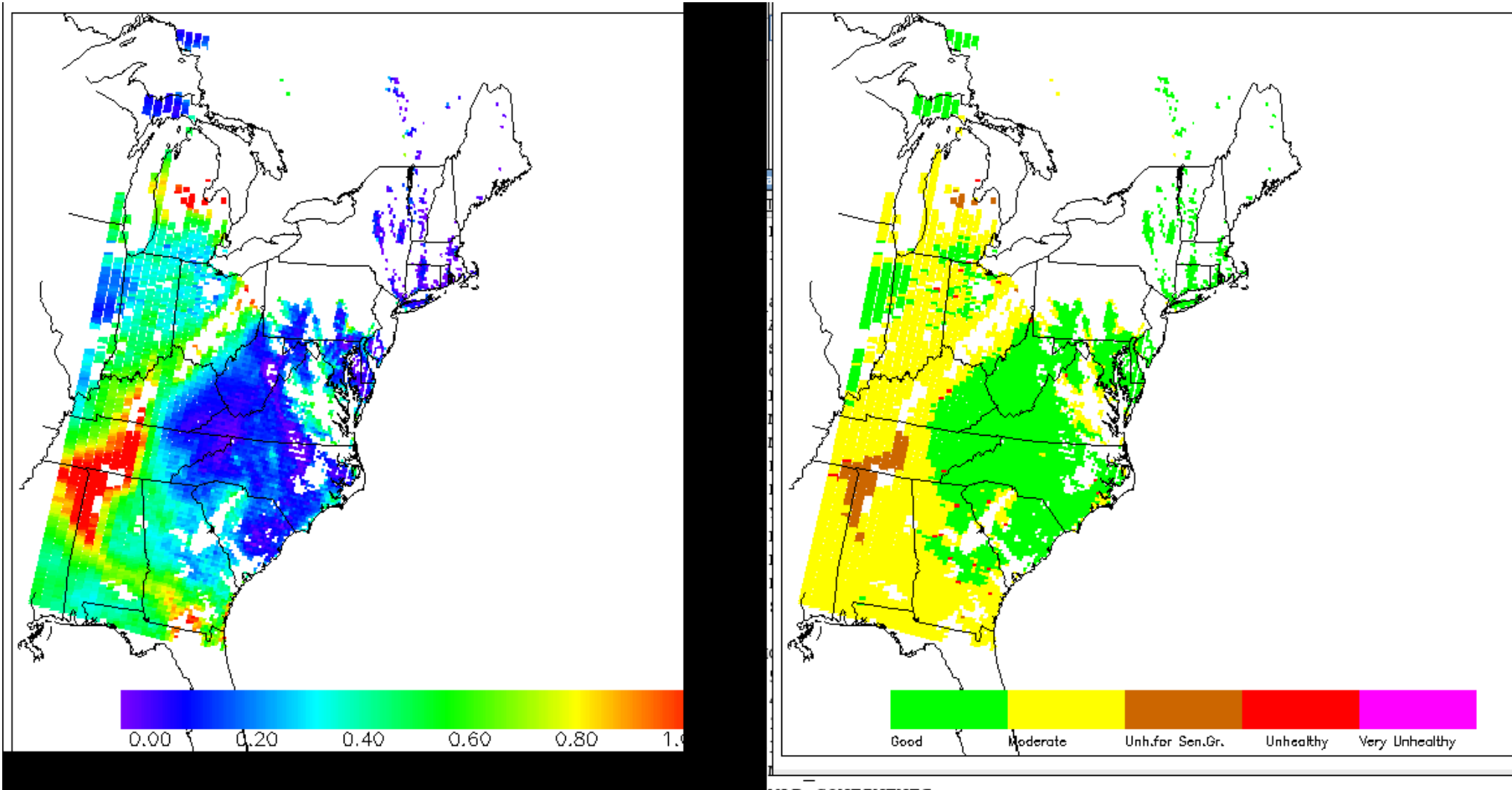
<https://airnow.gov/index.cfm?action=airnow.calculator>

```
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 m.drawcoastlines(linewidth=0.5)
129 m.drawparallels(np.arange(-90., 120., 5.), labels=[1, 0, 0, 0])
130 m.drawmeridians(np.arange(-180., 181., 5.), labels=[0, 0, 0, 1])
131 x, y = m(longitude, latitude)
132 my_cmap=LinearSegmentedColormap.from_list('mycmap', ['green','yellow','orange','red','purple','brown'],6
133 m.pcolormesh(x, y, data,cmap=my_cmap)
134 plt.clim(0,6)
135 #create colorbar
136 cb = m.colorbar()
137 cb.set_label('AQI Category')
138 cb.set_ticks([1.5, 1.5,2.5,3.5,4.5,5.5]) # force there to be only 7 ticks
139 cb.set_ticklabels(['Good', 'Moderate', 'Unhealthy for \nSensitive Groups','Unhealthy','Very Unhealthy','']
140
```

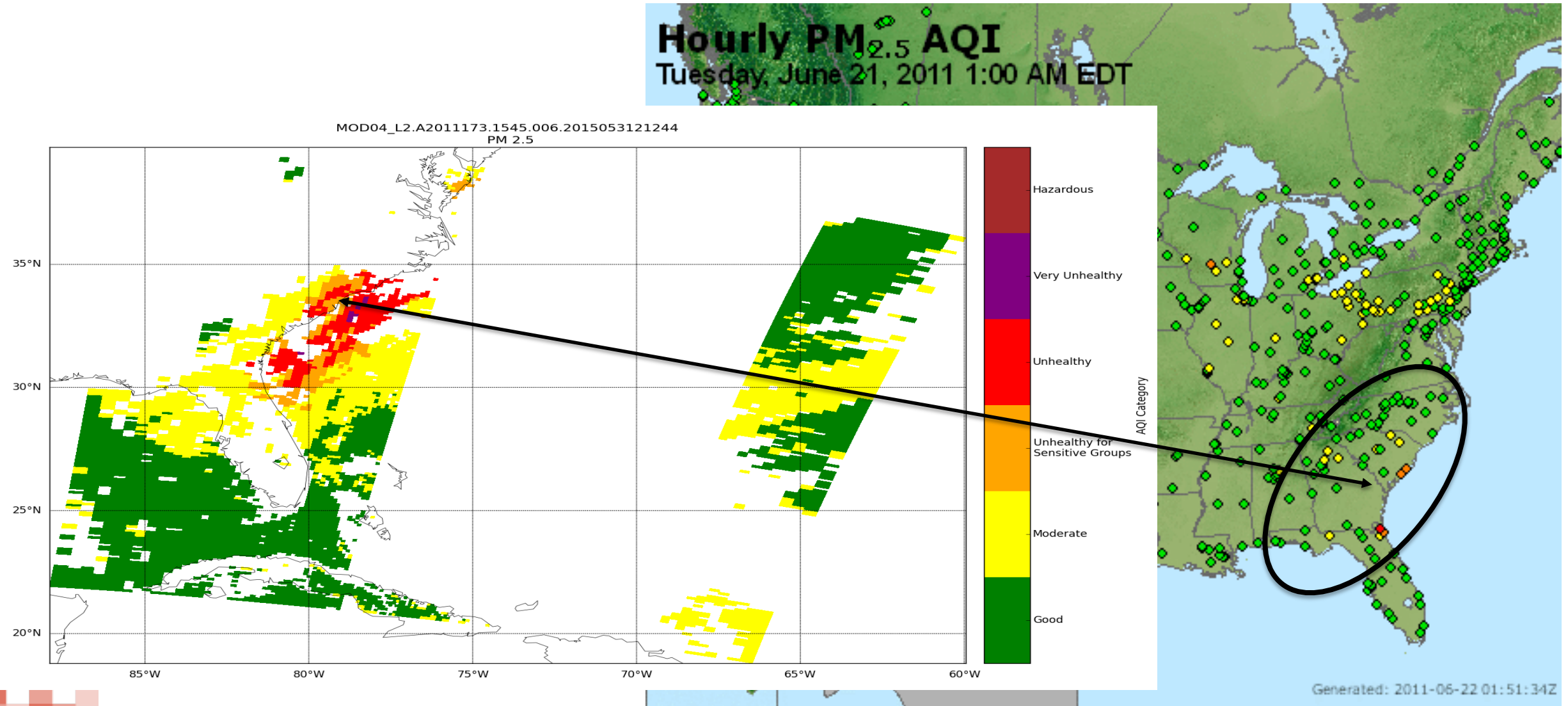
Change the
default slope &
intercept

Change the air
quality categories

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

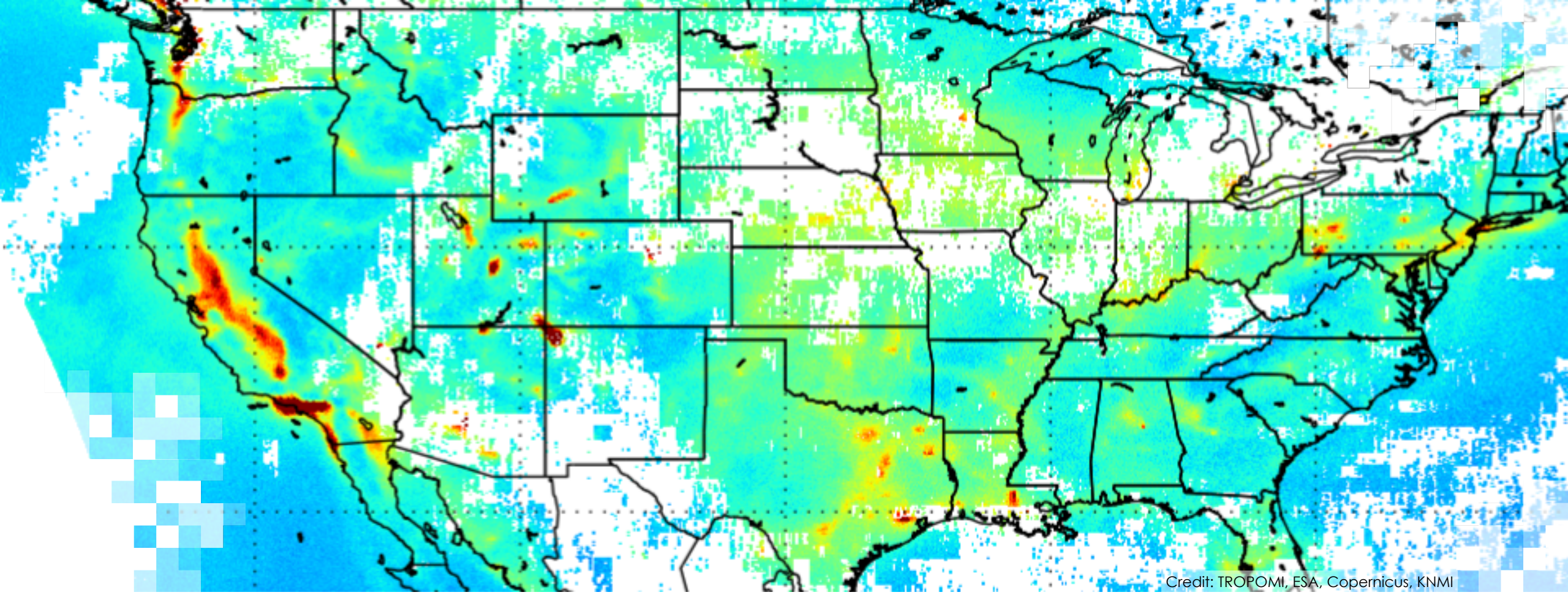


Application – Compare Satellite with Surface Maps



References

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



Questions?