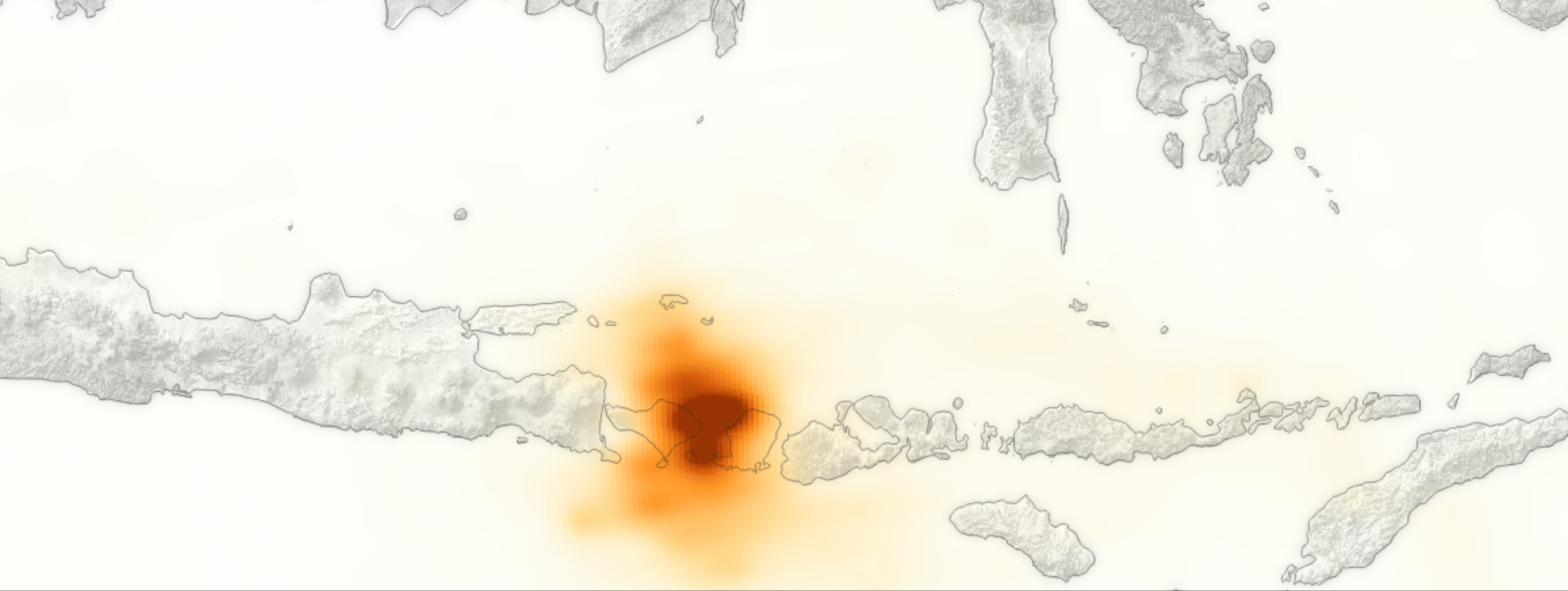


Read, Map, and Extract Level 2 OMI NO₂ and SO₂

Data Analysis Tools for High Resolution Air Quality Satellite Datasets

Pawan Gupta & Melanie Follette-Cook, January 17-22, 2018



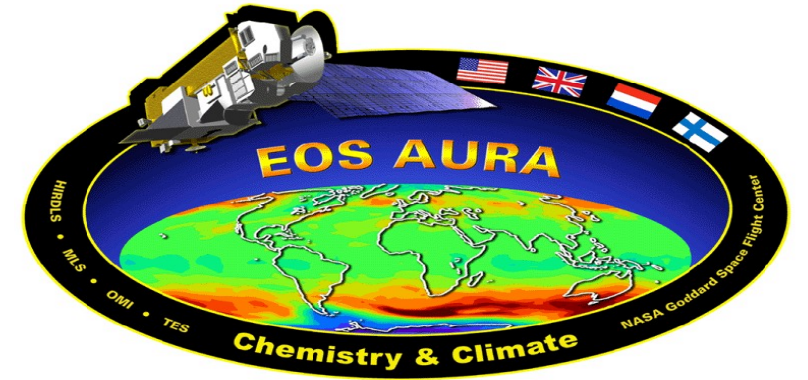


Ozone Monitoring Instrument (OMI)

Ozone Monitoring Instrument (OMI)

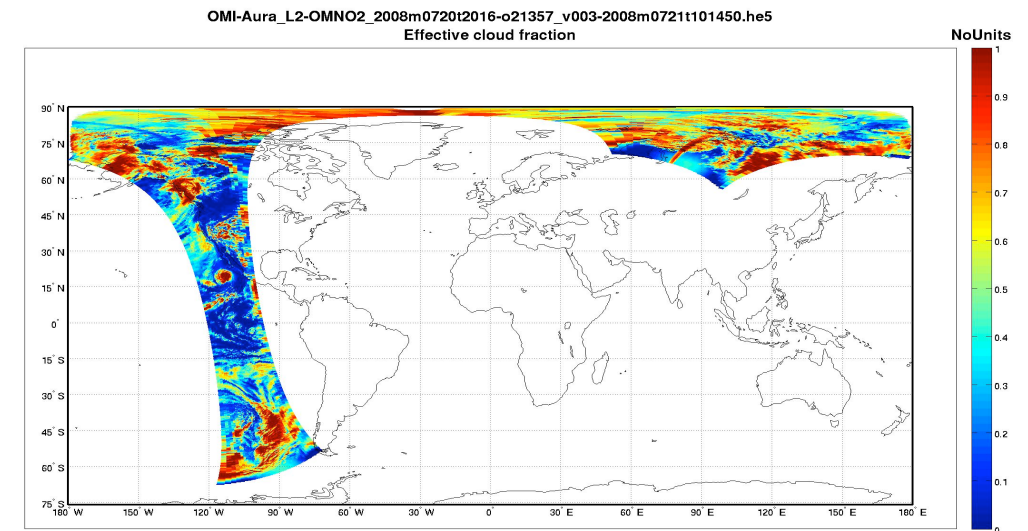
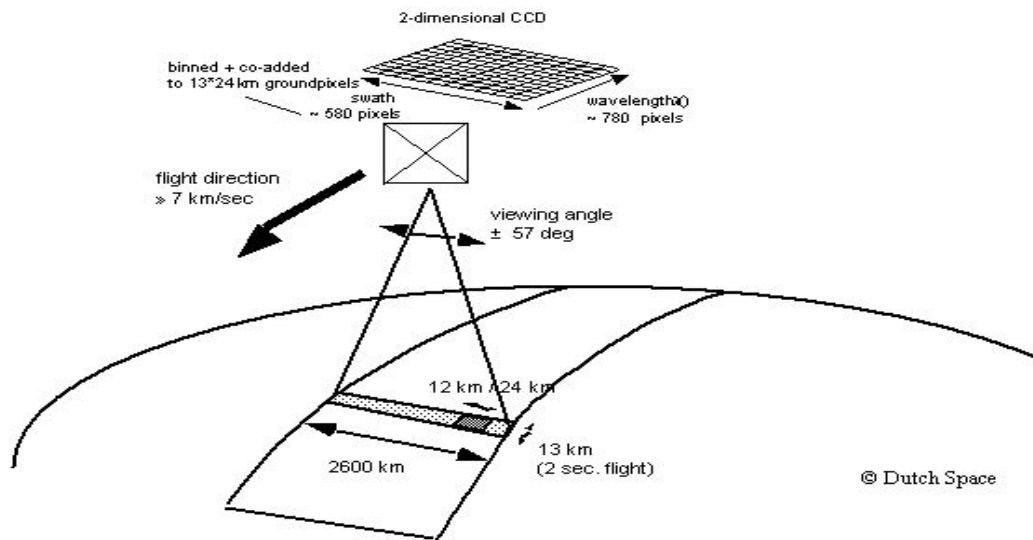
- Launched July 15, 2004
- NASA EOS Aura Satellite
- Nadir-viewing UV/Visible
 - 270 – 310 nm at 0.6 nm
 - 310 – 500 nm at 0.45 nm
- 1:45 p.m. equatorial crossing time
- 13x24 km² at nadir
- Daily global coverage

- Products
 - Total Column O₃
 - Tropospheric Column O₃
 - Aerosol optical depth (in UV)
 - Column Formaldehyde
 - Column NO₂
 - Tropospheric column NO₂
 - Column SO₂



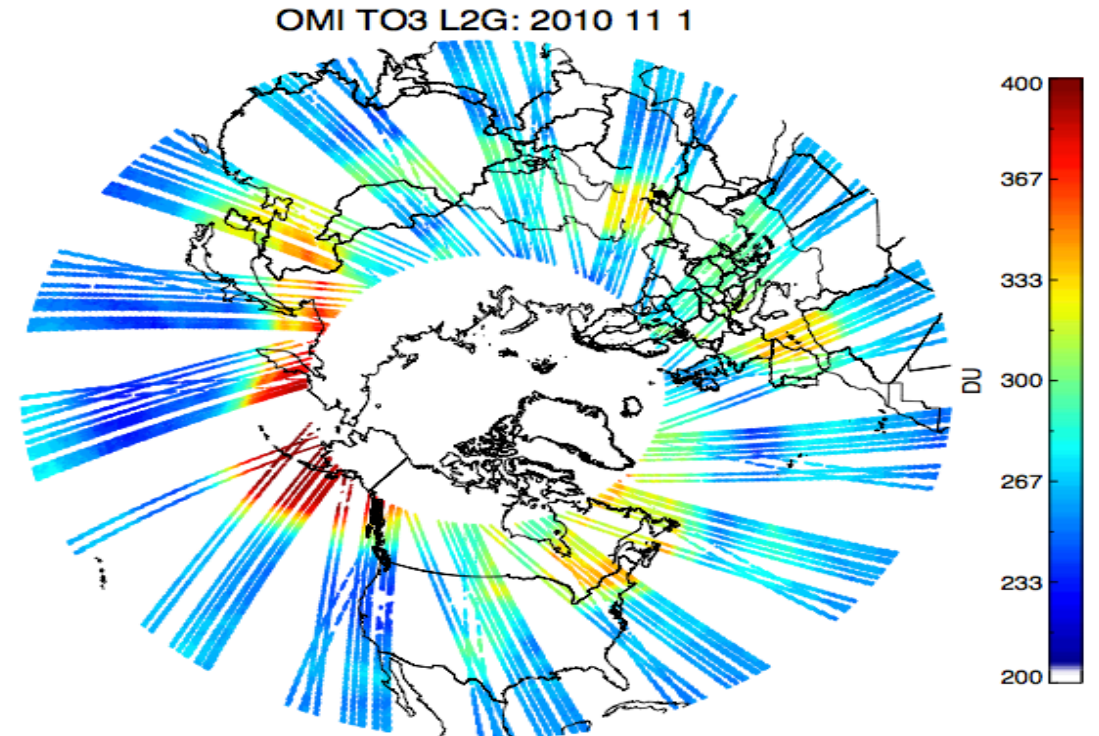
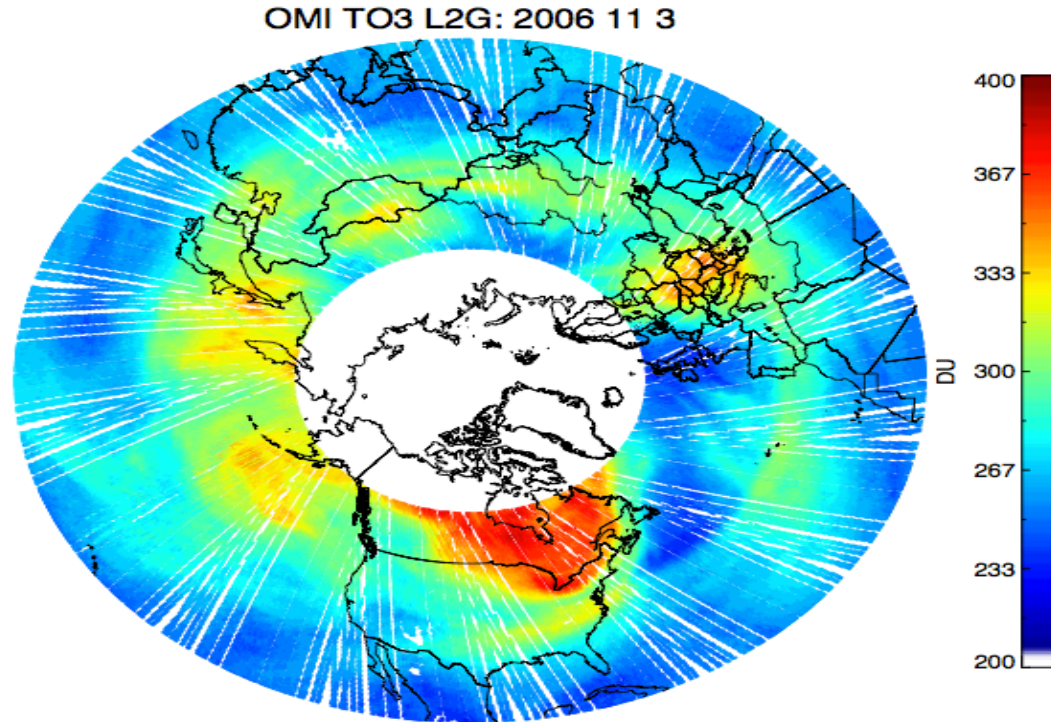
Data Granule

- Product File
 - covers sunlit portion of the orbit with an approx. 2,600 km wide swath
 - contains 60 binned pixels or scenes per viewing line
- 14 or 15 granules are produced daily, providing fully contiguous coverage of the globe



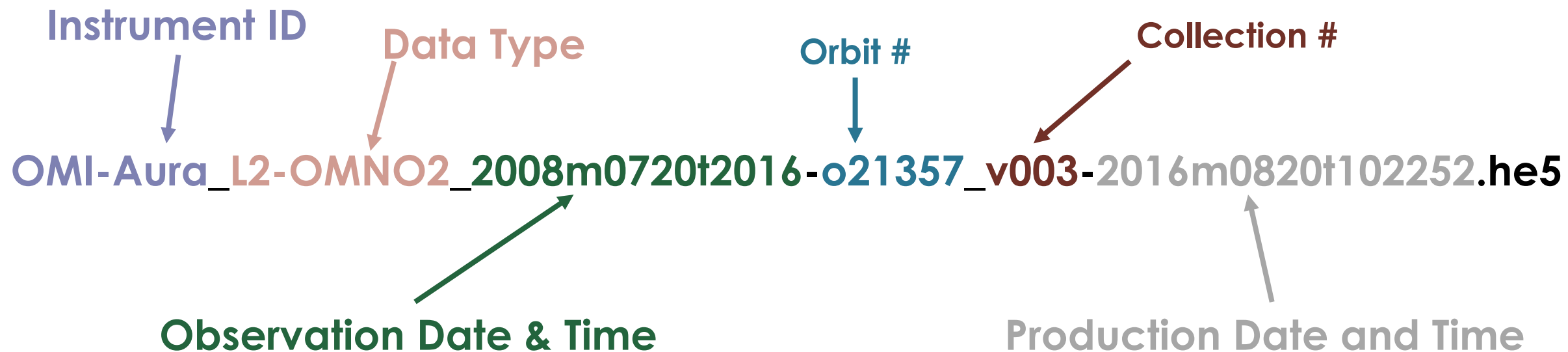
Important Information Regarding OMI

- Almost 50% data loss since 2008 (row anomaly effect)
- Affects all OMI products



Understanding an OMI File Name

OMNO2, OMNO2

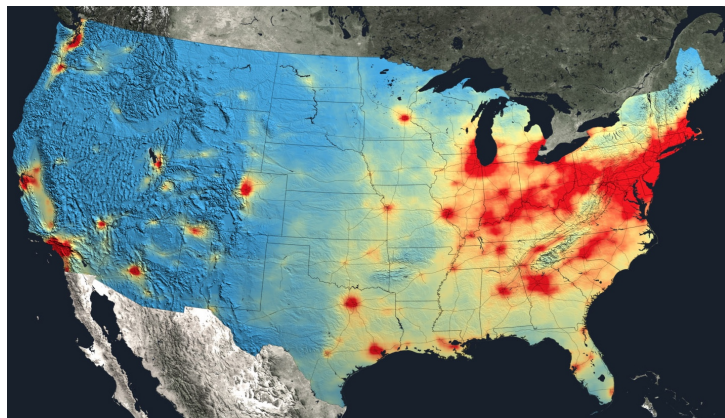


HDFLook, Panoply, IDL, Python, Fortran, MatLab, and more can be used to read the data



Nitrogen Dioxide (NO₂)

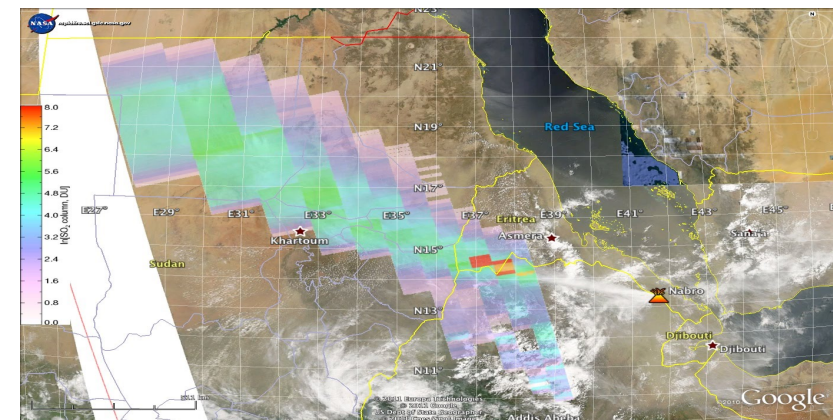
- EPA criteria pollutant, ozone precursor and health irritant
- Sources: Fires, industrial and transportation sources, stationary sources (e.g. power plants), lightning
- High concentrations in the planetary boundary layer (PBL) make tropospheric column amounts more useful for estimating surface levels



airquality.gsfc.nasa.gov

Sulfur Dioxide (SO₂)

- EPA Criteria pollutant
- Contributes to acid deposition and linked to adverse health effects
- Sources: Volcanoes, coal and oil burning



Aqua MODIS visible image of the Nabro (Eritrea) eruption on June 13, 2011 and the SO₂ plume overlaid.



Quantification of Gas Abundances - Units

Satellite Tracer	Units
OMI O ₃ , SO ₂	Dobson Units (DU)
OMI NO ₂	Molecules/cm ²

$$1 \text{ DU} = 2.69 \times 10^{16} \text{ molec/cm}^2$$

OMI NO₂ Parameter (SDS) information (OMNO2)

SDS name	Description	Unit	Notes
ColumnAmountNO2Trop	Tropospheric Column NO ₂	Molec / cm ²	<ul style="list-style-type: none"> Use only rows 4-54 (where the first row = 0) Use only scenes with: radiative cloud fraction < 0.5 solar zenith angle < 85° terrain reflectivity < 0.3
TerrainReflectivity		Unitless	Scale factor: 0.001
CloudRadianceFraction		Unitless	Scale factor: 0.001
SolarZenithAngle		Deg	In geolocation fields

- All fill values are high negative numbers: $(-2.100 \approx -1.26765 \times 10^{30})$

OMI SO₂ Level-2 Product Summary (OMSO2)

SO ₂ Product	SDS Name	Estimated center of plume	Use
PBL SO ₂	ColumnAmountSO2_PBL	0.9 km	Near-surface pollution
TRL SO ₂	ColumnAmountSO2_TRL	3 km	Volcanic degassing
TRM SO ₂	ColumnAmountSO2_TRM	8 km	Plumes from moderate eruptions, and long range pollution transport
STL SO ₂	ColumnAmountSO2_STL	18 km	Explosive volcanic eruptions

Note: Each retrieval listed here yields total column values, and represents a different assumption of SO₂ plume height. These should therefore not be added together.

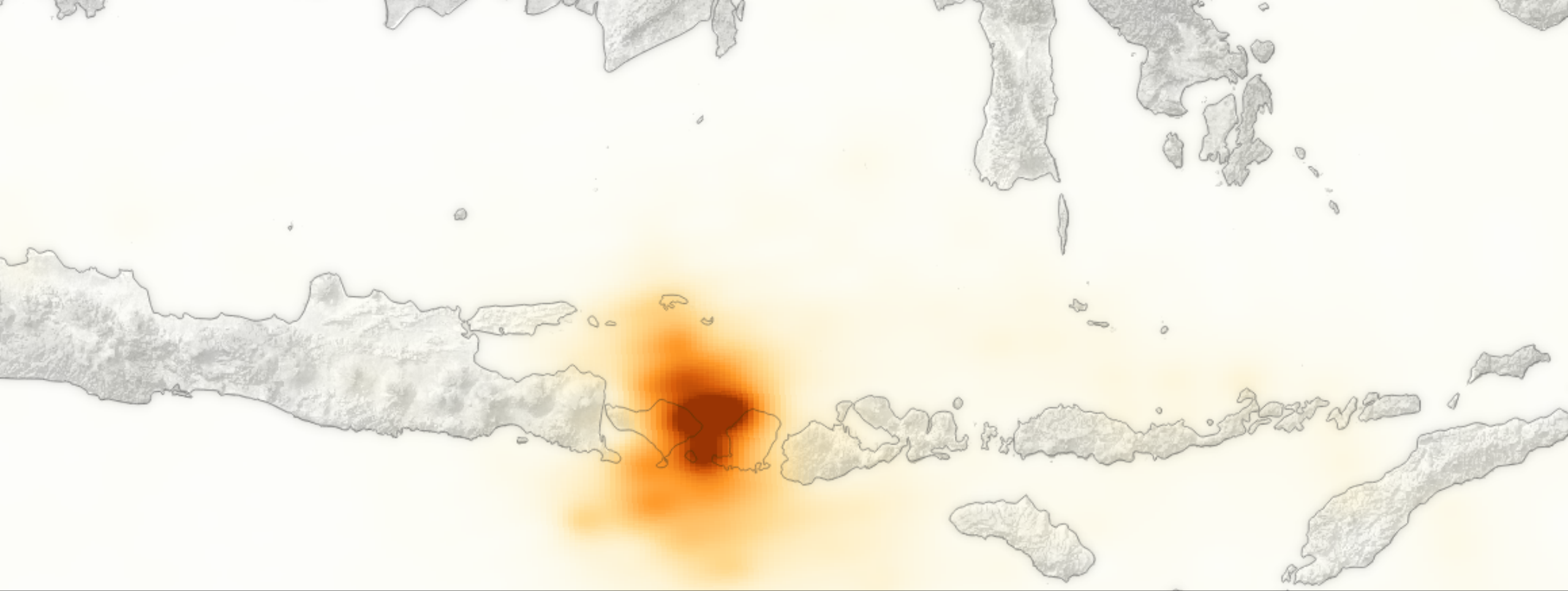


OMI SO₂ Parameter (SDS) information (OMSO2)

SDS name	Description	Unit	Notes
ColumnAmountSO2_PBL	Total Column SO2	DU	<ul style="list-style-type: none"> use only rows 4-54 (where the first row = 0) use only scenes with radiative cloud fraction < 0.3 solar zenith angle < 70°
ColumnAmountSO2_TRL /TRM/STL	Total Column SO2	DU	<ul style="list-style-type: none"> All rows can be used Use only scenes with solar zenith angle < 70°
RadiativeCloudFraction		Unitless	No scale factor
SolarZenithAngle		Deg	In geolocation fields

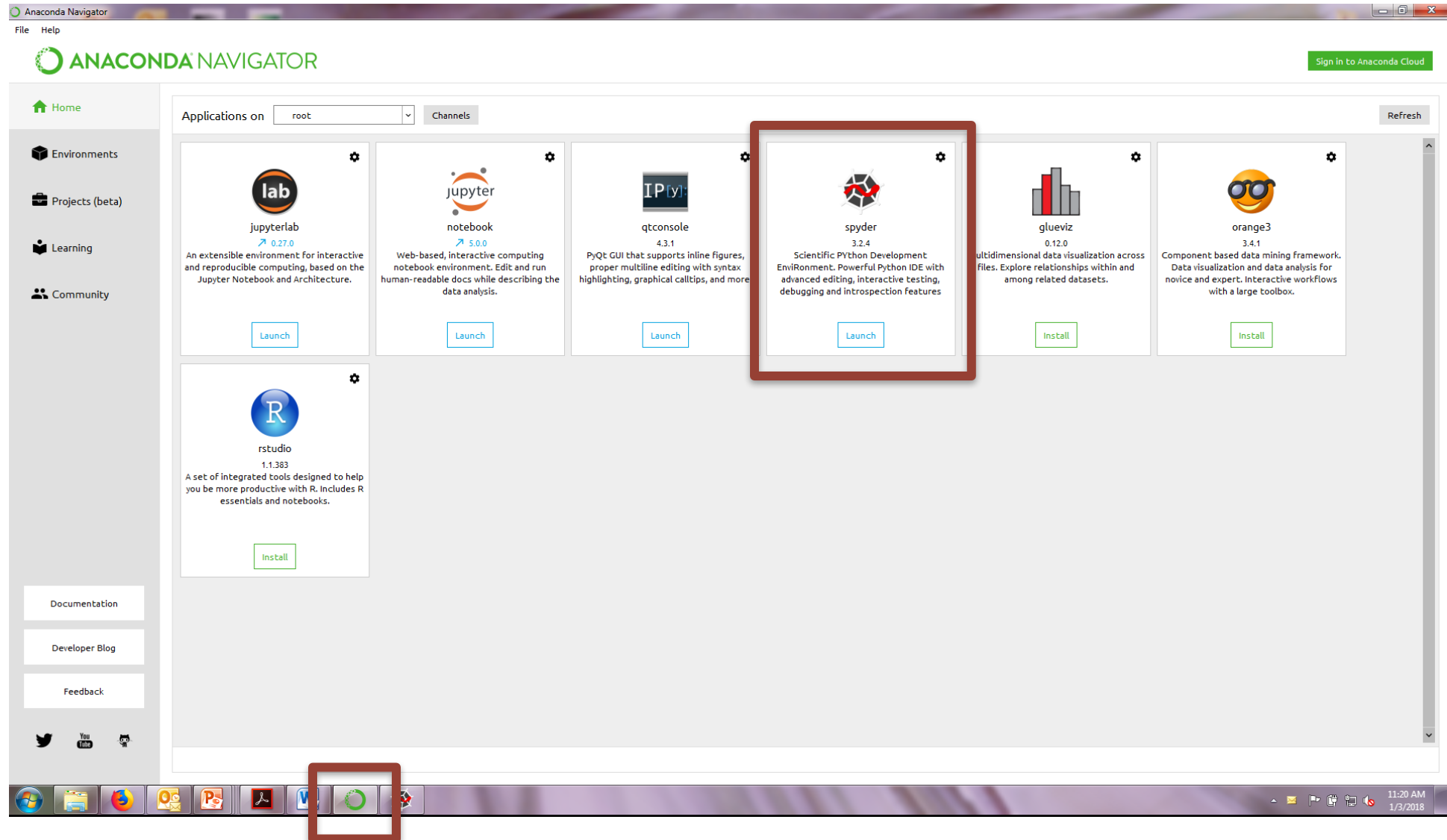
- As of the latest version (v1.3), the OMISO2 documentation does not recommend using the included data quality flags for screening
- All fill values are high negative numbers: $(-2.100 \approx -1.26765 \times 10^{30})$



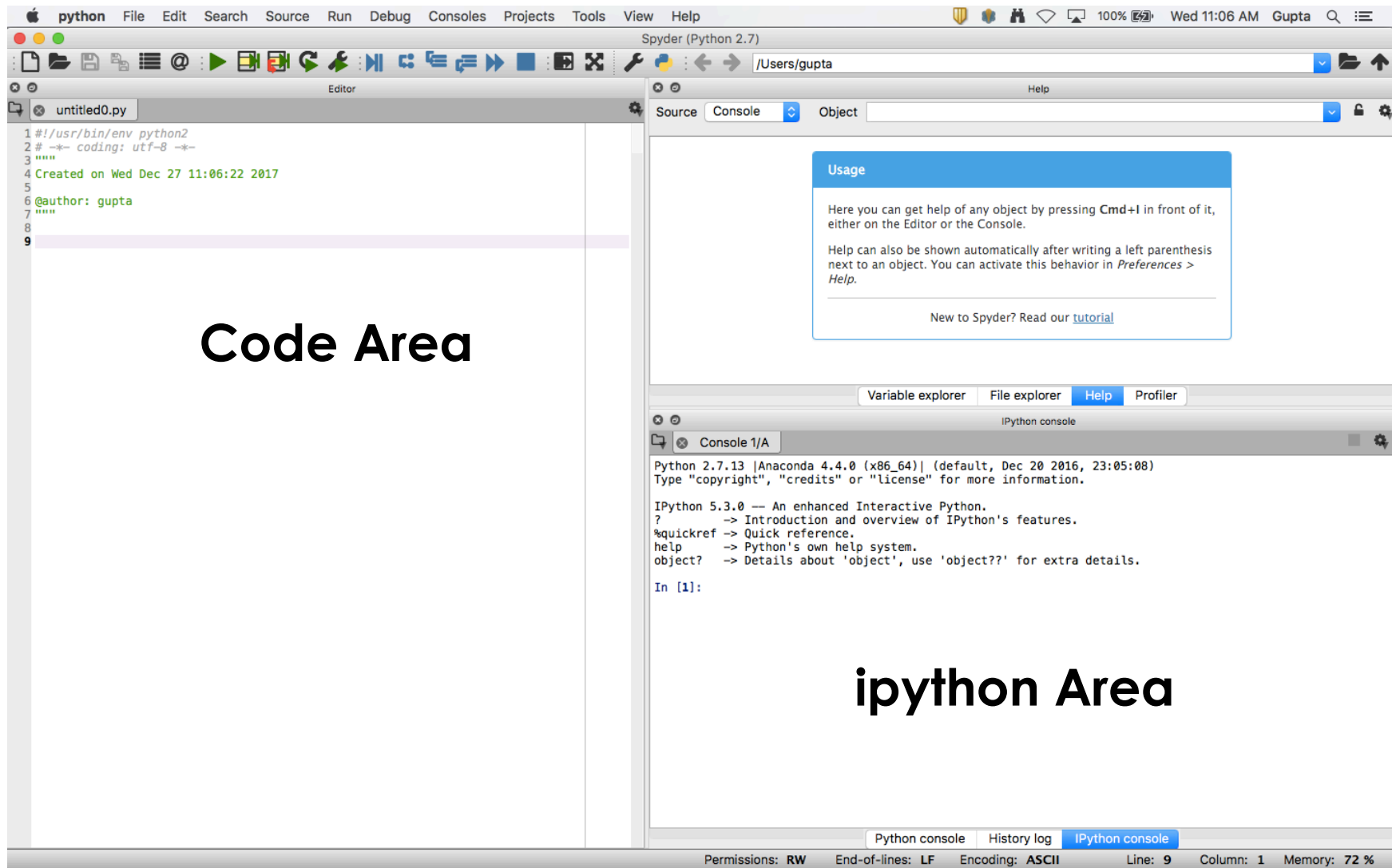


Getting Ready with Python

Anaconda & Spyder Editor

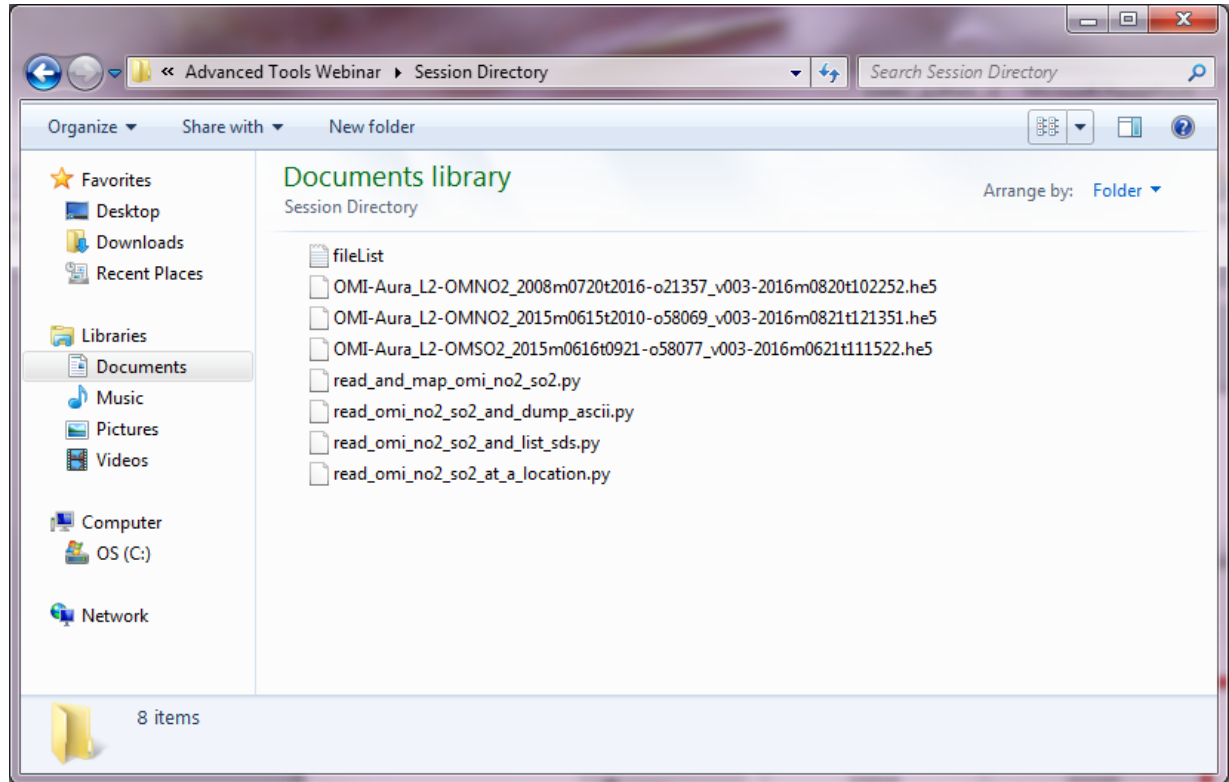


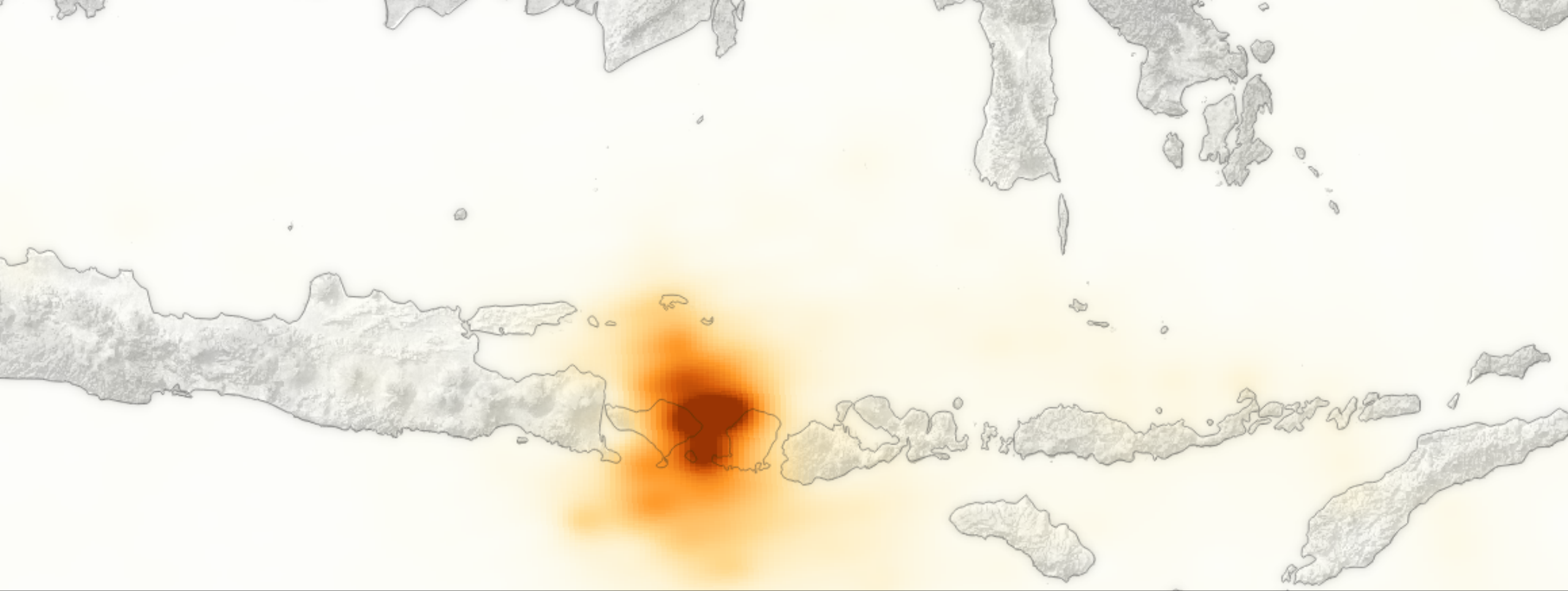
Spyder View



Current Directory View & fileList.txt

- In a text file, create a list of each HDF file of interest and name it, 'fileList.txt'
- The same directory should have
 - All the python codes
 - All the HDF data files
 - A file named 'fileList.txt' that contains a list of each HDF filename





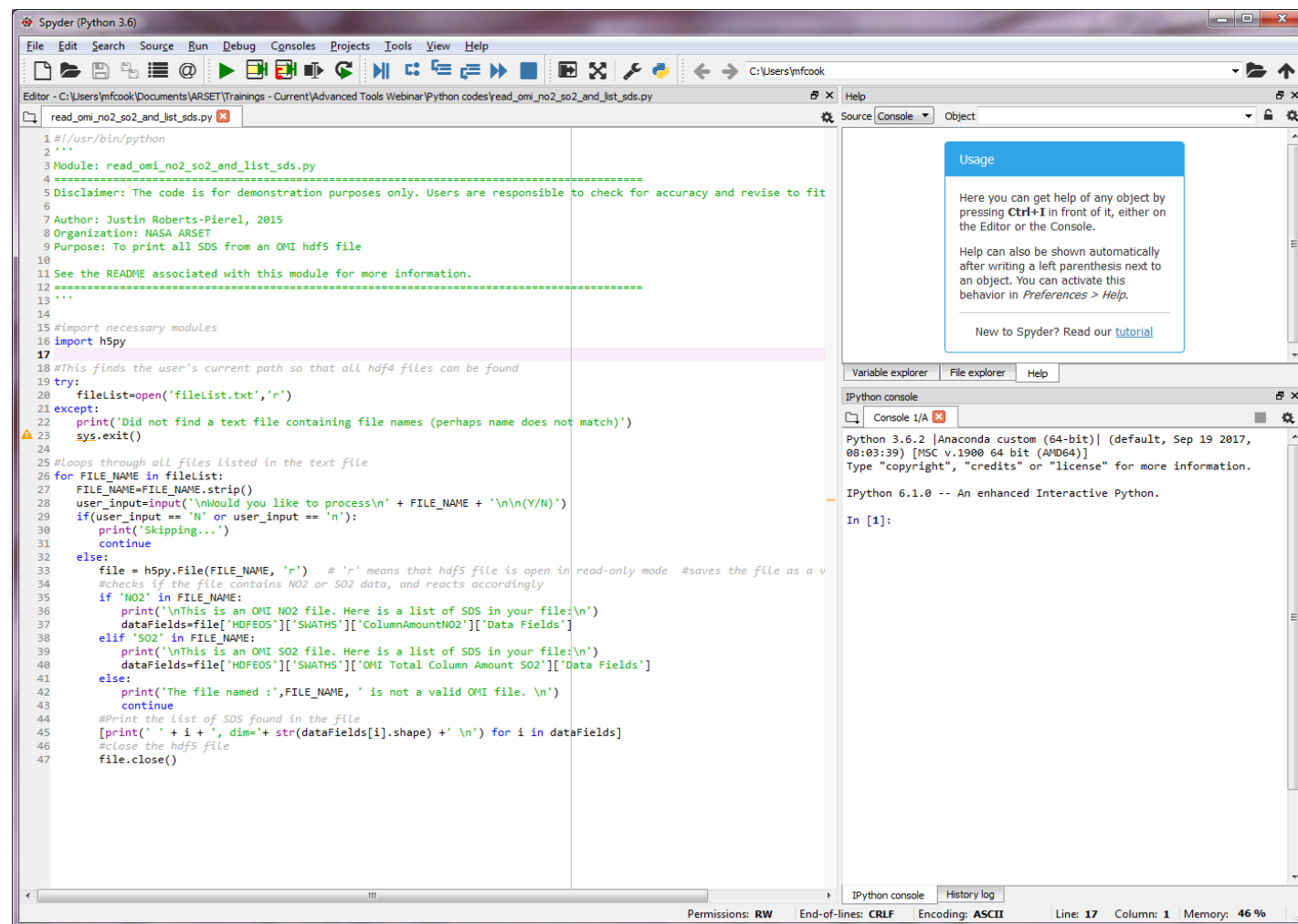
Read an OMI NO₂ File (HDF) and Print SDS List

Print Scientific Data Sets (SDSs)

read_omi_no2_so2_and_list_sds.py

Purpose: read OMI NO₂ or SO₂ level 2 data files in HDF format and print all the **Scientific Data Sets (SDS)**

In their current form, all of these codes work for *only level 2 products, not gridded products*



```
1 #!/usr/bin/python
2 '''
3 Module: read_omi_no2_so2_and_list_sds.py
4 =====
5 Disclaimer: The code is for demonstration purposes only. Users are responsible to check for accuracy and revise to fit
6
7 Author: Justin Roberts-Pierel, 2015
8 Organization: NASA ARSET
9 Purpose: To print all SDS from an OMI hdf5 file
10
11 See the README associated with this module for more information.
12 =====
13 '''
14
15 #import necessary modules
16 import h5py
17
18 #This finds the user's current path so that all hdf4 files can be found
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 match)')
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('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
29     if(user_input == 'N' or user_input == 'n'):
30         print('Skipping...')
31         continue
32     else:
33         file = h5py.File(FILE_NAME, 'r') # 'r' means that hdf5 file is open in read-only mode #saves the file as a v
34         #checks if the file contains NO2 or SO2 data, and reacts accordingly
35         if 'NO2' in FILE_NAME:
36             print('\nThis is an OMI NO2 file. Here is a list of SDS in your file:\n')
37             dataFields=file['HDFEOS']['SWATHS']['ColumnAmountNO2']['Data Fields']
38         elif 'SO2' in FILE_NAME:
39             print('\nThis is an OMI SO2 file. Here is a list of SDS in your file:\n')
40             dataFields=file['HDFEOS']['SWATHS']['OMI Total Column Amount SO2']['Data Fields']
41         else:
42             print('The file named :,FILE_NAME, ' is not a valid OMI file. \n')
43             continue
44         #Print the list of SDS found in the file
45         [print(' ' + i + ', dim='+ str(dataFields[i].shape) + ' \n') for i in dataFields]
46         #close the hdf5 file
47         file.close()
```

Usage

Here you can get help of any object by pressing **Ctrl+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

IPython console

Console I/A

Python 3.6.2 |Anaconda custom (64-bit)| (default, Sep 19 2017, 08:03:39) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 6.1.0 -- An enhanced Interactive Python.

In [1]:

Permissions: RW End-of-lines: CRLF Encoding: ASCII Line: 17 Column: 1 Memory: 46 %



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. The main editor window displays a Python script named `read_omi_no2_so2_and_list_sds.py`. A red circle highlights the green play button (run icon) in the toolbar. The code in the editor includes a disclaimer, author information, and a loop that processes files listed in `fileList.txt`. The console window on the right shows the output of the script, which is a list of data fields with their dimensions. The word "output" is written in large black text over the console output. The console also shows a prompt for user input: "Would you like to process OMI-Aura_L2-OMNO2_2015m0615t2010-o58069_v003-2016m0821t121351.he5 (Y/N)".

```
1 #!/usr/bin/python
2 ...
3 Module: read_omi_no2_so2_and_list_sds.py
4 -----
5 Disclaimer: The code is for demonstration purposes only. Users are responsible to check for accuracy and revise to fit their objective.
6
7 Author: Justin Roberts-Piere1, 2015
8 Organization: NASA ARSET
9 Purpose: To print all SDS from an OMI hdf5 file
10
11 See the README associated with this module for more information.
12 -----
13 '''
14
15 #import necessary modules
16 import h5py
17
18 #This finds the user's current path so that all hdf4 files can be found
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 match)')
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('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
29     if(user_input == 'N' or user_input == 'n'):
30         print('Skipping...')
31         continue
32     else:
33         file = h5py.File(FILE_NAME, 'r') # 'r' means that hdf5 file is open in read-only mode #saves the file as a variable named 'hdf'
34         #checks if the file contains NO2 or SO2 data, and reacts accordingly
35         if 'NO2' in FILE_NAME:
36             print('\nThis is an OMI NO2 file. Here is a list of SDS in your file:\n')
37             dataFields=file['HDFEOS']['SWATHS']['ColumnAmountNO2']['Data Fields']
38         elif 'SO2' in FILE_NAME:
39             print('\nThis is an OMI SO2 file. Here is a list of SDS in your file:\n')
40             dataFields=file['HDFEOS']['SWATHS']['OMI Total Column Amount SO2']['Data Fields']
41         else:
42             print('The file named:',FILE_NAME, ' is not a valid OMI file. \n')
43             continue
44         #Print the List of SDS found in the file
45         [print(' ' + i + ', dim='+ str(dataFields[i].shape) + ' \n') for i in dataFields]
46         #Close the hdf5 file
47         file.close()
```

Usage

Here you can get help of any object by pressing **Ctrl+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

IPython console

Console I/A

SlantColumnAmountNO2Std, dim=(1644, 60)

SmallPixelRadiance, dim=(10, 60)

SmallPixelRadiancePointer, dim=(1644, 2)

TerrainHeight, dim=(1644, 60)

TerrainPressure, dim=(1644, 60)

TerrainReflectivity, dim=(1644, 60)

TropopausePressure, dim=(1644, 60)

VcdApBelowCloud, dim=(1644, 60)

VcdApStrat, dim=(1644, 60)

VcdApTrop, dim=(1644, 60)

VcdQualityFlags, dim=(1644, 60)

WavelengthRegistrationCheck, dim=(1644, 60)

WavelengthRegistrationCheckStd, dim=(1644, 60)

XTrackQualityFlags, dim=(1644, 60)

Would you like to process
OMI-Aura_L2-OMNO2_2015m0615t2010-o58069_v003-2016m0821t121351.he5
(Y/N)

Permissions: RW End-of-lines: CRLF Encoding: ASCII Line: 17 Column: 1 Memory: 46 %



Editing the Code

```
1 #!/usr/bin/python
2 '''
3 Module: read_omi_no2_so2_and_list_sds.py
4 =====
5 Disclaimer: The code is for demonstration purposes only. Users are responsible to check for accuracy and revise to fit their objective.
6
7 Author: Justin Roberts-Pierel, 2015
8 Organization: NASA ARSET
9 Purpose: To print all SDS from an OMI hdf5 file
10
11 See the README associated with this module for more information.
12 =====
13 '''
14
15 #import necessary modules
16 import h5py
17
18 #This finds the user's current path so that all hdf4 files can be found
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 match)')
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('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
29     if(user_input == 'N' or user_input == 'n'):
30         print('Skipping...')
31         continue
32     else:
33         file = h5py.File(FILE_NAME, 'r') # 'r' means that hdf5 file is open in read-only mode #saves the file as a v
34         #checks if the file contains NO2 or SO2 data, and reacts accordingly
35         if 'NO2' in FILE_NAME:
36             print('\nThis is an OMI NO2 file. Here is a list of SDS in your file:\n')
37             dataFields=file['HDFEOS']['SWATHS']['ColumnAmountNO2']['Data Fields']
38         elif 'SO2' in FILE_NAME:
39             print('\nThis is an OMI SO2 file. Here is a list of SDS in your file:\n')
40             dataFields=file['HDFEOS']['SWATHS']['OMI Total Column Amount SO2']['Data Fields']
41         else:
42             print('The file named :,FILE_NAME, ' is not a valid OMI file. \n')
43             continue
44         #Print the list of SDS found in the file
45         [print(' ' + i + ', dim='+ str(dataFields[i].shape) + ' \n') for i in dataFields]
46         #close the hdf5 file
47         file.close()
```

Change the file to be read

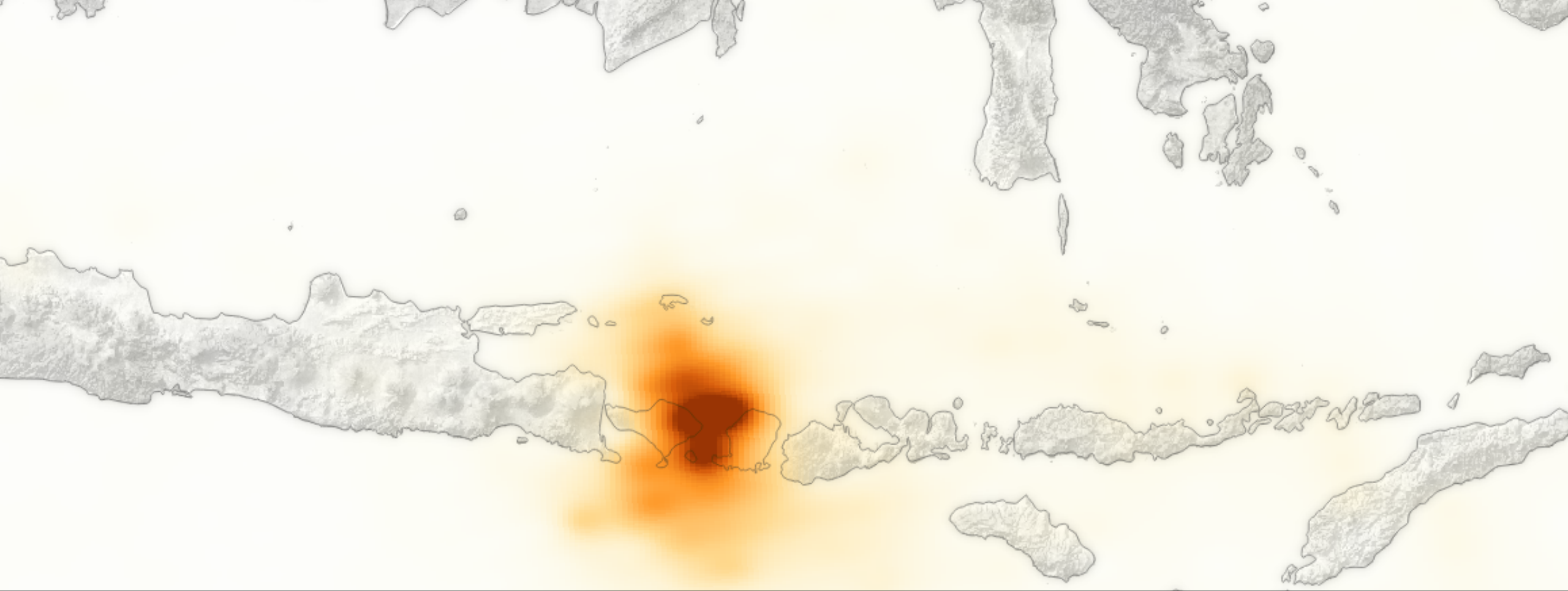
By changing the location of dataFields to geolocation (found in other codes) this can also list the available geolocation variables



Applications

- OMI Level 2 NO₂ and SO₂ data are provided in the HDF file
- Each HDF file contains several geophysical parameters
- Special codes and tools are required to open the HDF files
- This code helps users see the names and dimensions of the available SDSs inside an HDF file for further analysis

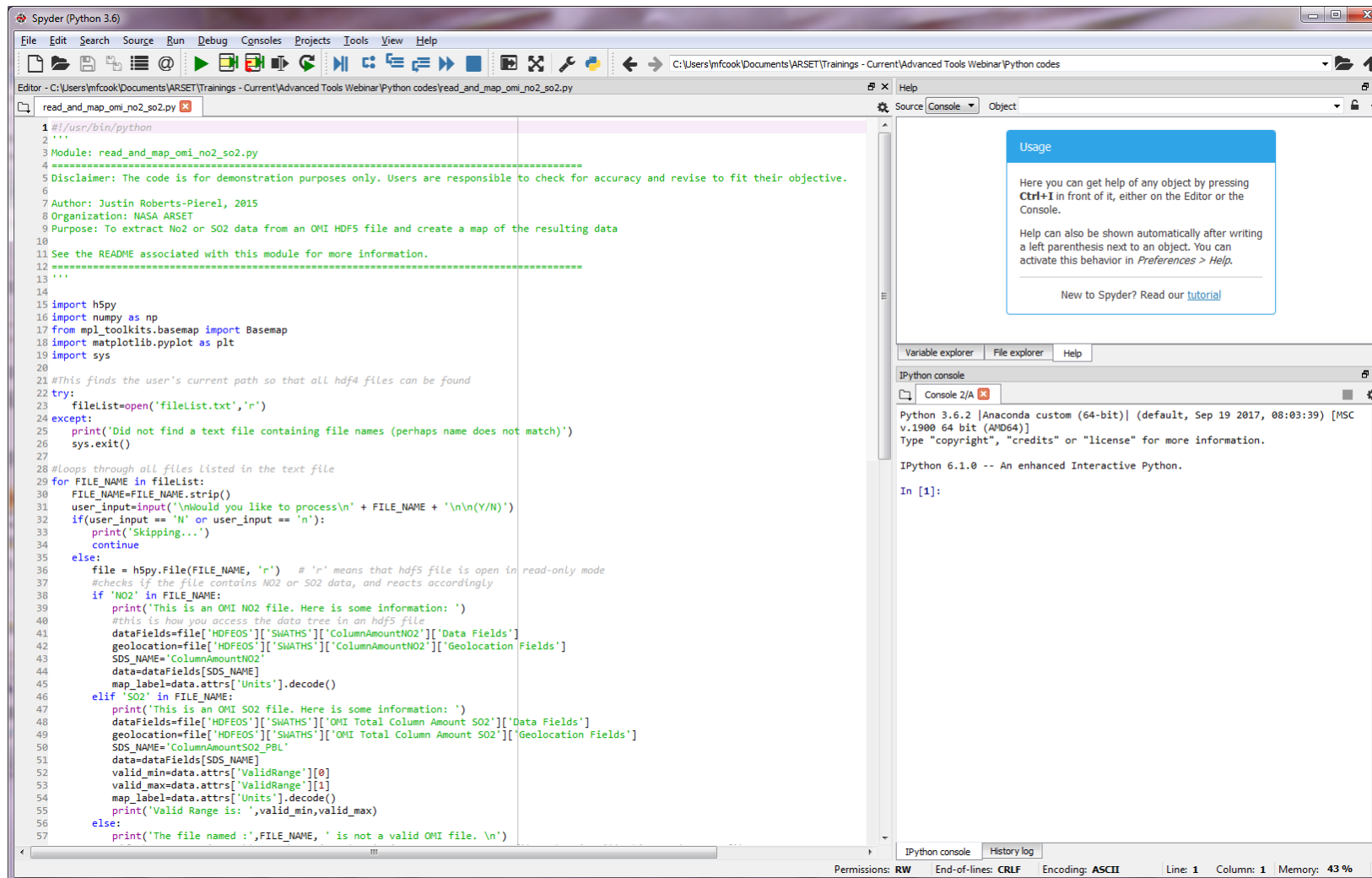




Map NO_2 or SO_2

Plot and Save a Map of OMI NO₂ or SO₂

read_and_map_omi_so2_no2.py



```
1#!/usr/bin/python
2'''
3Module: read_and_map_omi_no2_so2.py
4-----
5Disclaimer: The code is for demonstration purposes only. Users are responsible to check for accuracy and revise to fit their objective.
6
7Author: Justin Roberts-Pierel, 2015
8Organization: NASA ARSET
9Purpose: To extract No2 or SO2 data from an OMI HDF5 file and create a map of the resulting data
10
11See the README associated with this module for more information.
12-----
13'''
14
15import h5py
16import numpy as np
17from mpl_toolkits.basemap import Basemap
18import matplotlib.pyplot as plt
19import sys
20
21#This finds the user's current path so that all hdf4 files can be found
22try:
23    fileList=open('filelist.txt','r')
24except:
25    print('Did not find a text file containing file names (perhaps name does not match)')
26    sys.exit()
27
28#Loops through all files listed in the text file
29for FILE_NAME in fileList:
30    FILE_NAME=FILE_NAME.strip()
31    user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
32    if(user_input == 'N' or user_input == 'n'):
33        print('Skipping...')
34        continue
35    else:
36        file = h5py.File(FILE_NAME, 'r') # 'r' means that hdf5 file is open in read-only mode
37        #checks if the file contains NO2 or SO2 data, and reacts accordingly
38        if 'NO2' in FILE_NAME:
39            print('This is an OMI NO2 file. Here is some information: ')
40            #this is how you access the data tree in an hdf5 file
41            dataFields=file['HDFEOS']['SWATHS']['ColumnAmountNO2']['Data Fields']
42            geolocation=file['HDFEOS']['SWATHS']['ColumnAmountNO2']['Geolocation Fields']
43            SDS_NAME='ColumnAmountNO2'
44            data=dataFields[SDS_NAME]
45            map_label=data.attrs['Units'].decode()
46        elif 'SO2' in FILE_NAME:
47            print('This is an OMI SO2 file. Here is some information: ')
48            dataFields=file['HDFEOS']['SWATHS']['OMI Total Column Amount SO2']['Data Fields']
49            geolocation=file['HDFEOS']['SWATHS']['OMI Total Column Amount SO2']['Geolocation Fields']
50            SDS_NAME='ColumnAmountSO2_PBL'
51            data=dataFields[SDS_NAME]
52            valid_min=data.attrs['ValidRange'][0]
53            valid_max=data.attrs['ValidRange'][1]
54            map_label=data.attrs['Units'].decode()
55            print('Valid Range is: ',valid_min,valid_max)
56        else:
57            print('The file named :',FILE_NAME, ' is not a valid OMI file. \n')
```



Running and Output

```
In [1]: runfile('C:/Users/mfcook/Documents/ARSET/Trainings - Current/Advanced Tools Webinar/Python codes/  
read_and_map_omi_no2_so2.py', wdir='C:/Users/mfcook/Documents/ARSET/Trainings - Current/Advanced Tools Webinar/  
Python codes')
```

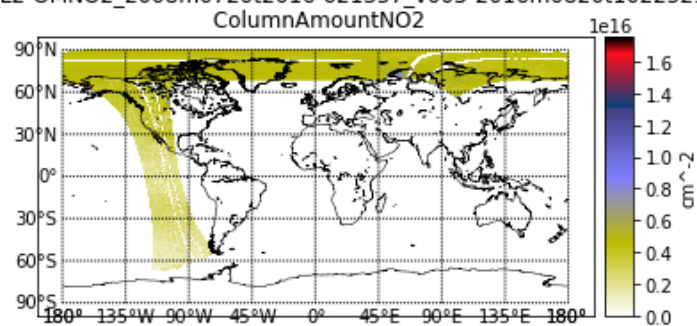
```
Would you like to process  
OMI-Aura_L2-OMNO2_2008m0720t2016-o21357_v003-2016m0820t102252.he5
```

```
(Y/N)Y
```

```
This is an OMI NO2 file. Here is some information:  
3.14792e+15  
The average of this data is: 3.14792e+15  
The standard deviation is: 1.35182e+15  
The median is: 2.90004e+15  
The range of latitude in this file is: -75.0061 to 89.8693 degrees  
The range of longitude in this file is: -179.99 to 179.975 degrees
```

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

```
OMI-Aura_L2-OMNO2_2008m0720t2016-o21357_v003-2016m0820t102252.he5
```



Output Map

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

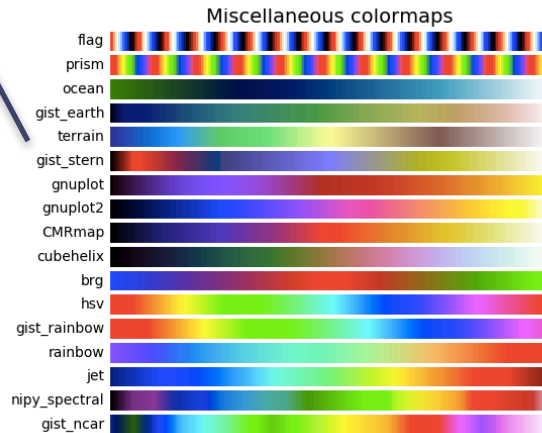
NO₂/SO₂
Statistics



Editing the Code

Change the color scale

```
93 if is_map == 'Y' or is_map == 'y':
94     data = np.ma.masked_array(data, np.isnan(data))
95     m = Basemap(projection='cyl', resolution='l',
96                 llcrnrlat=-90, urcrnrlat = 90,
97                 llcrnrlon=-180, urcrnrlon = 180)
98     m.drawcoastlines(linewidth=0.5)
99     m.drawparallels(np.arange(-90., 120., 30.), labels=[1, 0, 0, 0])
100    m.drawmeridians(np.arange(-180, 180., 45.), labels=[0, 0, 0, 1])
101    my_cmap = plt.cm.get_cmap('gist_stern_r')
102    my_cmap.set_under('w')
103    m.pcolormesh(lon, lat, data, latlon=True, vmin=0, vmax=np.nanmax(dat
104    cb = m.colorbar()
105    cb.set_label(map_label)
106    plt.autoscale()
107    #title the plot
108    plt.title('{0}\n {1}'.format(FILE_NAME,
109    fig = plt.gcf()
110    # Show the plot window.
111    plt.show()
112    #once you close the map it asks if you'd l
113    is_save=str(input('\nWould you like to
114    if is_save == 'Y' or is_save == 'y':
115        #saves as a png if the user would l
116        pngfile = '{0}.png'.format(FILE_NAMI
117        fig.savefig(pngfile)
118    #close the hdf5 file
119    file.close()
```



Change the SDS to plot

```
28 #Loops through all files listed in the text file
29 for FILE_NAME in fileList:
30     FILE_NAME=FILE_NAME.strip()
31     user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
32     if(user_input == 'N' or user_input == 'n'):
33         print('Skipping...')
34         continue
35     else:
36         file = h5py.File(FILE_NAME, 'r') # 'r' means that hdf5 file is open in read-only mode
37         #checks if the file contains NO2 or SO2 data, and reacts accordingly
38         if 'NO2' in FILE_NAME:
39             print('This is an OMI NO2 file. Here is some information: ')
40             #this is how you access the data tree in an hdf5 file
41             dataFields=file['HDFEOS']['SWATHS']['ColumnAmountNO2']['Data Fields']
42             geolocation=file['HDFEOS']['SWATHS']['ColumnAmountNO2']['Geolocation Fields']
43             SDS_NAME='ColumnAmountNO2'
44             data=dataFields[SDS_NAME]
45             map_label=data.attrs['Units'].decode()
46         elif 'SO2' in FILE_NAME:
47             print('This is an OMI SO2 file. Here is some information: ')
48             dataFields=file['HDFEOS']['SWATHS']['OMI Total Column Amount SO2']['Data Fields']
49             geolocation=file['HDFEOS']['SWATHS']['OMI Total Column Amount SO2']['Geolocation Fields']
50             SDS_NAME='ColumnAmountSO2_PBL'
51             data=dataFields[SDS_NAME]
52             valid_min=data.attrs['ValidRange'][0]
53             valid_max=data.attrs['ValidRange'][1]
54             map_label=data.attrs['Units'].decode()
55             print('Valid Range is: ',valid_min,valid_max)
56         else:
57             print('The file named: ',FILE_NAME, ' is not a valid OMI file. \n')
58             #if the program is unable to determine that it is an OMI SO2 or NO2 file, then it will skip
59             continue
```

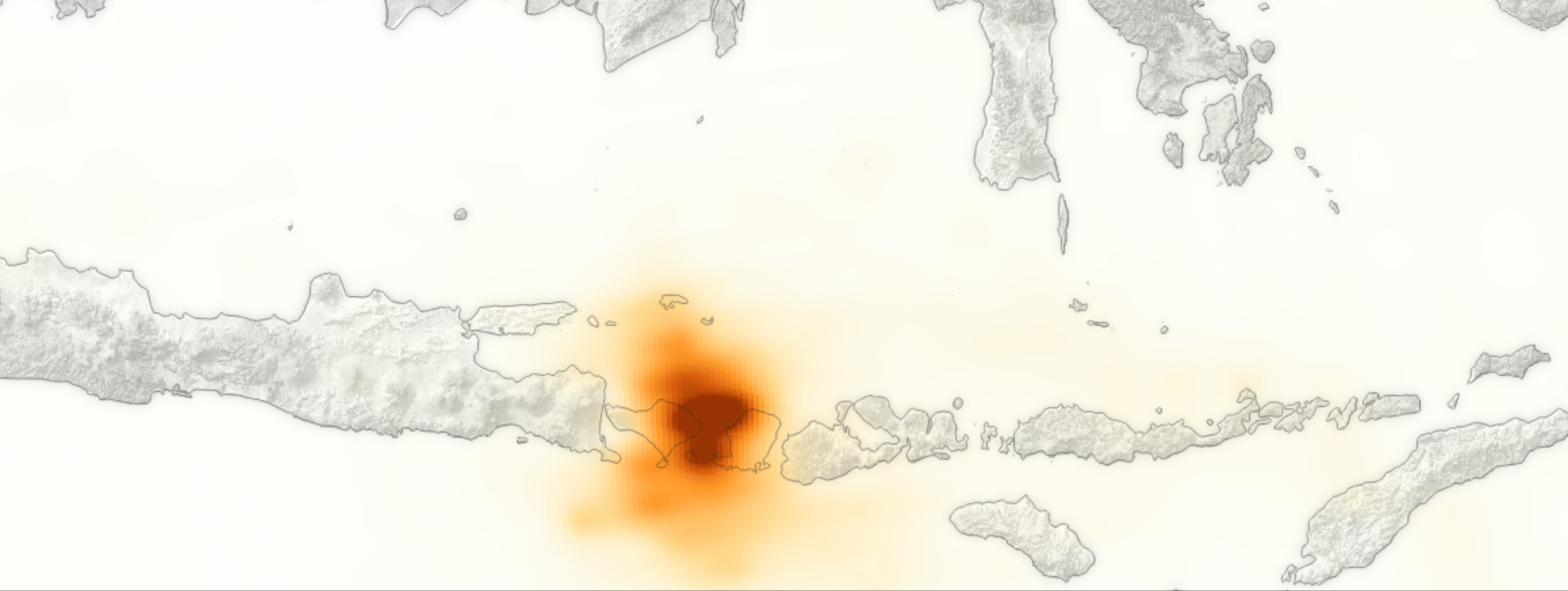
https://matplotlib.org/examples/color/colormaps_reference.html



Applications

- This is a sample code to read and map the OMI Level 2 NO₂ and SO₂ data
- The code can be modified to address various mapping needs
- User can create daily maps of trace gas columns over certain regions and start analyzing changes over time
- These maps can also help identify regions of high pollution





Extract NO_2/SO_2 at a given location

Extract Level 2 OMI NO₂/SO₂ Values

read_omi_no2_so2_at_a_location.py

- **Purpose:** read an OMI NO₂/SO₂ level 2 data file in HDF format and extract values at a given ground location

```
1 #!/usr/bin/python
2 ...
3 Module: read_omi_no2_so2_at_a_location.py
4 =====
5 Disclaimer: The code is for demonstration purposes only. Users are responsible to check for accuracy and revise to
6
7 Author: Justin Roberts-Pierel, 2015
8 Organization: NASA ARSET
9 Purpose: To view info about a variety of SDS from an OMI h5 file both generally and at a specific lat/lon
10
11 See the README associated with this module for more information.
12 =====
13 ...
14
15 #import necessary modules
16 import h5py
17 import numpy as np
18 import sys
19 from numpy import unravel_index
20
21 #This finds the user's current path so that all hdf4 files can be found
22 try:
23     fileList=open('fileList.txt','r')
24 except:
25     print('Did not find a text file containing file names (perhaps name does not match)')
26     sys.exit()
27
28 #loops through all files listed in the text file
29 for FILE_NAME in fileList:
30     FILE_NAME=FILE_NAME.strip()
31     user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
32     if(user_input == 'N' or user_input == 'n'):
33         print('Skipping...')
34         continue
35     else:
36         file = h5py.File(FILE_NAME, 'r') # 'r' means that hdf5 file is open in read-only mode
37         if 'NO2' in FILE_NAME:
38             print('This is an OMI NO2 file. Here is some information: ')
39             #this is how you access the data tree in an hdf5 file
40             dataFields=file['HDFEOS']['SMATHS']['ColumnAmountNO2']['Data Fields']
41             geolocation=file['HDFEOS']['SMATHS']['ColumnAmountNO2']['Geolocation Fields']
42             SDS_NAME='ColumnAmountNO2'
43             data=dataFields[SDS_NAME]
44             map_label=data.attrs['Units'].decode()
45         elif 'SO2' in FILE_NAME:
46             print('This is an OMI SO2 file. Here is some information: ')
47             dataFields=file['HDFEOS']['SMATHS']['OMI Total Column Amount SO2']['Data Fields']
48             geolocation=file['HDFEOS']['SMATHS']['OMI Total Column Amount SO2']['Geolocation Fields']
49             SDS_NAME='ColumnAmountSO2_PBL'
50             data=dataFields[SDS_NAME]
51             valid_min=data.attrs['ValidRange'][0]
52             valid_max=data.attrs['ValidRange'][1]
53             map_label=data.attrs['Units'].decode()
54             print('Valid Range is: ',valid_min,valid_max)
55         else:
56             print('The file named: ',FILE_NAME, ' is not a valid OMI file. \n')
57             #if the program is unable to determine that it is an OMI SO2 or NO2 file, then it will skip to the next
```

File explorer

Name	Size	Type	Date Modified
OMNO2_SDS_new.txt	1 KB	txt File	12/20/2017 10:30 AM
OMSO2_SDS_new.txt	1 KB	txt File	12/21/2017 4:11 PM
read_and_map_omi_no2_so2.py	4 KB	py File	9/9/2015 12:10 PM
read_omi_no2_so2_and_dump_ascii.py	5 KB	py File	12/22/2017 10:57 AM
read_omi_no2_so2_and_list_sds_geo.py	2 KB	py File	1/2/2018 3:56 PM

Python console

```
Python 3.6.2 [Anaconda custom (64-bit)] (default, Sep 19 2017, 08:03:39) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.

Python 6.1.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/mfcook/Documents/ARSET/Trainings - Current/Advanced Tools Webinar/Python codes/
read_omi_no2_so2_at_a_location.py', wdir='C:/Users/mfcook/Documents/ARSET/Trainings - Current/Advanced Tools Webinar/Python
codes')

Would you like to process
OMI-Aura_L2-OMNO2_2008m0720t2016-o21357_v003-2016m0820t102252.he5
(Y/N)
This is an OMI NO2 file. Here is some information:
The range of latitude in this file is: -75.0861 to 89.8693 degrees
The range of longitude in this file is: -179.99 to 179.975 degrees

Please enter the latitude you would like to analyze (Deg. N): 30
Please enter the longitude you would like to analyze (Deg. E): -100
855 59

The nearest pixel to your entered location is at:
Latitude: 29.8233 Longitude: -101.774
The value of ColumnAmountNO2 at this pixel is 3.92950208633e+15
There are 9 valid pixels in a 3x3 grid centered at your entered location.
The average value in this grid is: 4.15249517773e+15
The median value in this grid is: 4.01630659661e+15
The standard deviation in this grid is: 2.77808737236e+14

There are 25 valid pixels in a 5x5 grid centered at your entered location.
The average value in this grid is: 4.05478825804e+15
The median value in this grid is: 3.96426125666e+15
The standard deviation in this grid is: 4.40895029635e+14

Would you like to process
OMI-Aura_L2-OMNO2_2015m0615t2010-o58069_v003-2016m0821t121351.he5
(Y/N)
```

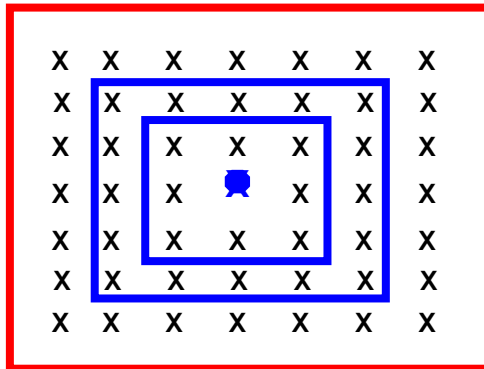


Running and Output

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

Latitude and
Longitude of the
Station

Outputs



```
Would you like to process  
OMI-Aura_L2-OMNO2_2008m0720t2016-o21357_v003-2016m0820t102252.he5  
  
(Y/N)Y  
This is an OMI NO2 file. Here is some information:  
The range of latitude in this file is: -75.0061 to 89.8693 degrees  
The range of longitude in this file is: -179.99 to 179.975 degrees
```

```
Please enter the latitude you would like to analyze (Deg. N): 30
```

```
Please enter the longitude you would like to analyze (Deg. E): -100  
855 59
```

```
The nearest pixel to your entered location is at:  
Latitude: 29.8233 Longitude: -101.774  
The value of ColumnAmountNO2 at this pixel is 3.92950208633e+15  
There are 9 valid pixels in a 3x3 grid centered at your entered location.  
The average value in this grid is: 4.15249517773e+15  
The median value in this grid is: 4.01630659661e+15  
The standard deviation in this grid is: 2.77808737236e+14
```

```
There are 25 valid pixels in a 5x5 grid centered at your entered location.
```

```
The average value in this grid is: 4.05478825804e+15  
The median value in this grid is: 3.96426125666e+15  
The standard deviation in this grid is: 4.40095029635e+14
```

```
Would you like to process  
OMI-Aura_L2-OMNO2_2015m0615t2010-o58069_v003-2016m0821t121351.he5
```

```
(Y/N)
```



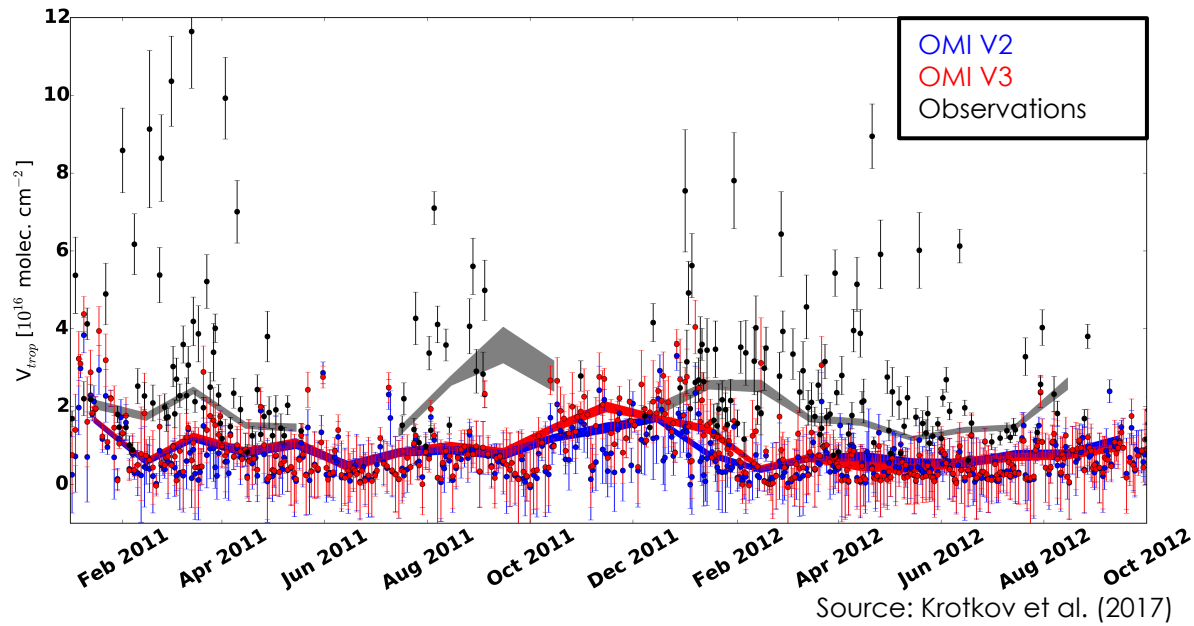
Editing the Code – Change the SDS

```
27
28 #Loops through all files listed in the text file
29 for FILE_NAME in fileList:
30     FILE_NAME=FILE_NAME.strip()
31     user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
32     if(user_input == 'N' or user_input == 'n'):
33         print('Skipping...')
34         continue
35     else:
36         file = h5py.File(FILE_NAME, 'r') # 'r' means that hdf5 file is open in read-only mode
37         if 'NO2' in FILE_NAME:
38             print('This is an OMI NO2 file. Here is some information: ')
39             #this is how you access the data tree in an hdf5 file
40             dataFields=file['HDFEOS']['SWATHS']['ColumnAmountNO2']['Data Fields']
41             geolocation=file['HDFEOS']['SWATHS']['ColumnAmountNO2']['Geolocation Fields']
42             SDS_NAME='ColumnAmountNO2'
43             data=dataFields[SDS_NAME]
44             map_label=data.attrs['Units'].decode()
45         elif 'SO2' in FILE_NAME:
46             print('This is an OMI SO2 file. Here is some information: ')
47             dataFields=file['HDFEOS']['SWATHS']['OMI Total Column Amount SO2']['Data Fields']
48             geolocation=file['HDFEOS']['SWATHS']['OMI Total Column Amount SO2']['Geolocation Fields']
49             SDS_NAME='ColumnAmountSO2_PBL'
50             data=dataFields[SDS_NAME]
51             valid_min=data.attrs['ValidRange'][0]
52             valid_max=data.attrs['ValidRange'][1]
53             map_label=data.attrs['Units'].decode()
54             print('Valid Range is: ',valid_min,valid_max)
55         else:
56             print('The file named :',FILE_NAME, ' is not a valid OMI file. \n')
57             #if the program is unable to determine that it is an OMI SO2 or NO2 file, then it will skip to the next file
58             continue
```

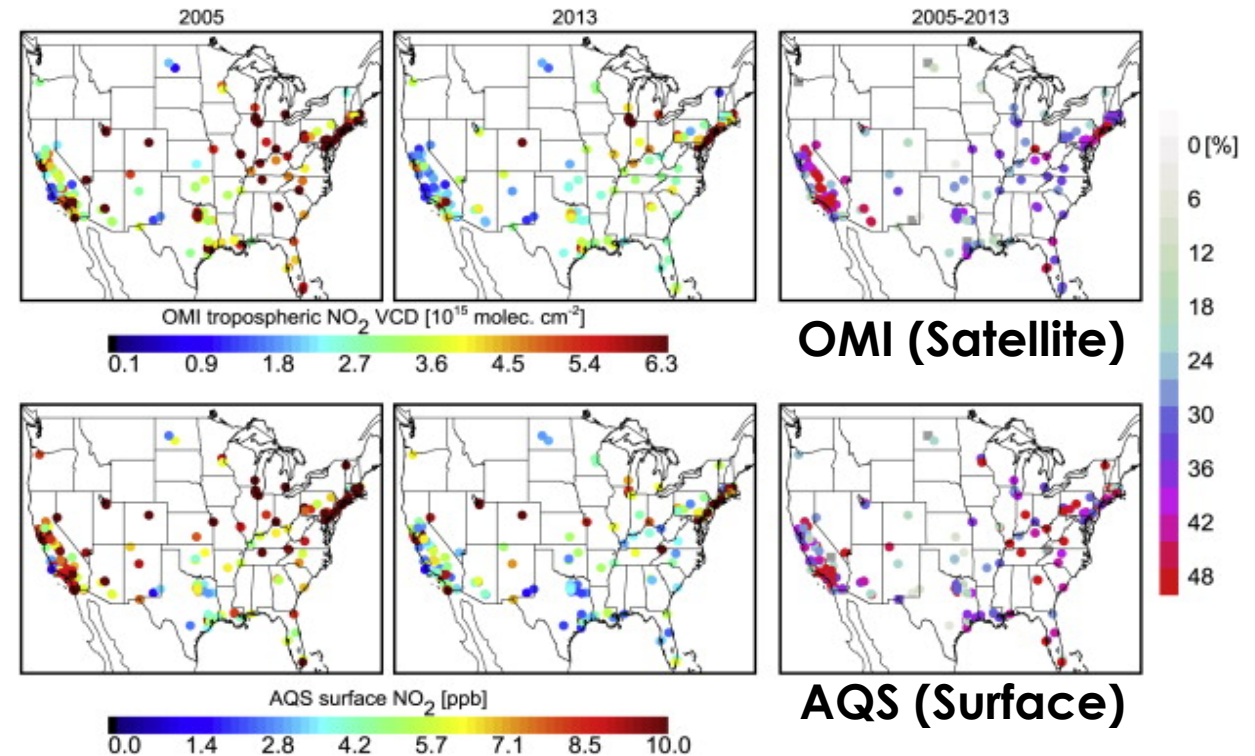


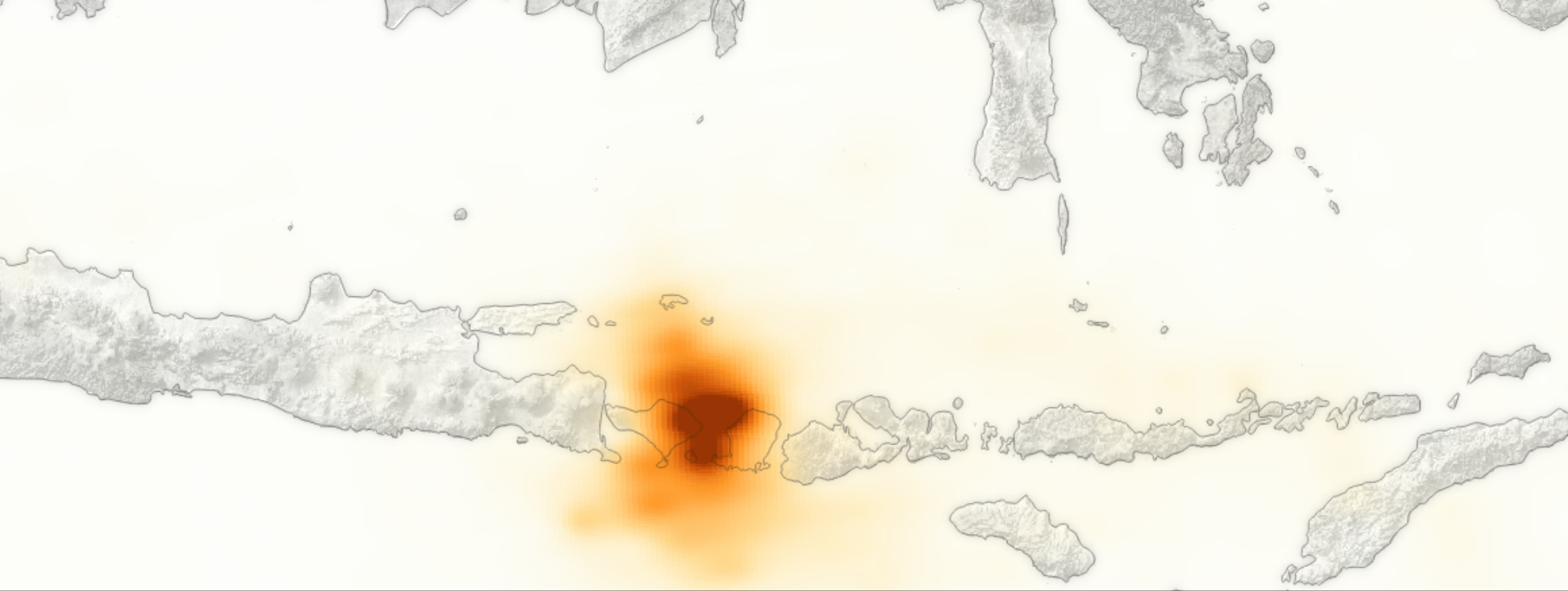
Applications

Satellite Validation



Column vs. Surface Relationship and Trends



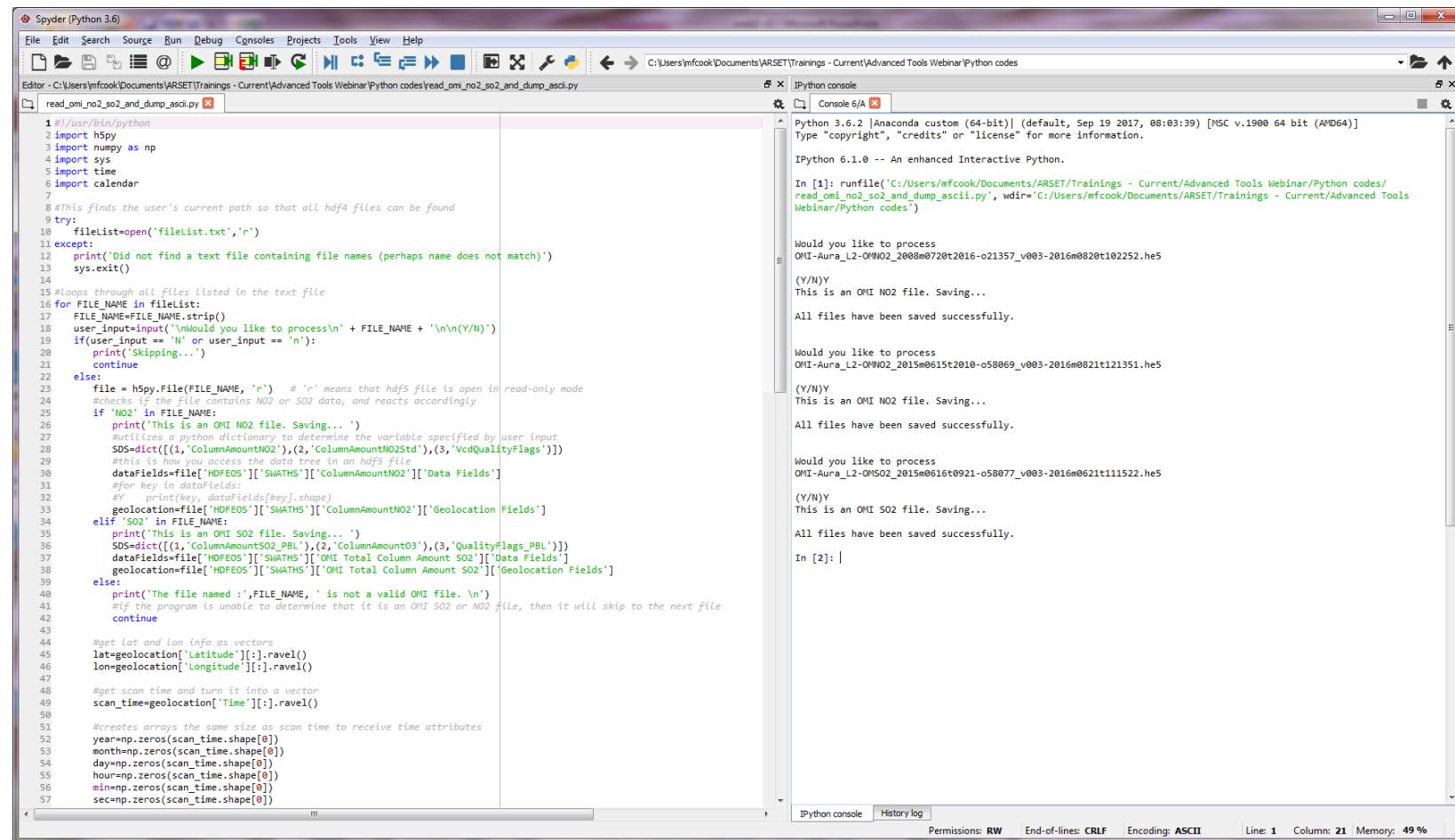


Output HDF variables to CSV

Output OMI NO2/SO2 HDF Variables to a CSV File

read_omi_no2_so2_and_dump_ascii.py

- **Purpose:** read an OMI level 2 NO₂ or SO₂ data file in HDF format and write certain SDSs into a csv (text) file



```
1 #!/usr/bin/python
2 import h5py
3 import numpy as np
4 import sys
5 import time
6 import calendar
7
8 #This finds the user's current path so that all hdf4 files can be found
9 try:
10     fileList=open('fileList.txt','r')
11 except:
12     print('Did not find a text file containing file names (perhaps name does not match)')
13     sys.exit()
14
15 #Loops through all files listed in the text file
16 for FILE_NAME in fileList:
17     FILE_NAME=FILE_NAME.strip()
18     user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
19     if(user_input == 'N' or user_input == 'n'):
20         print('Skipping...')
21         continue
22     else:
23         file = h5py.File(FILE_NAME, 'r') # 'r' means that hdf5 file is open in read-only mode
24         #checks if the file contains NO2 or SO2 data, and reacts accordingly
25         if 'NO2' in FILE_NAME:
26             print('This is an OMI NO2 file. Saving.. ')
27             #utilizes a python dictionary to determine the variable specified by user input
28             SDS=dict({'1','ColumnAmountNO2'},(2,'ColumnAmountNO2Std'),(3,'VcdQualityFlags'))
29             #this is how you access the data tree in an hdf5 file
30             dataFields=file['HDFEOS']['SWATHS']['ColumnAmountNO2']['Data Fields']
31             #for key in dataFields:
32                 #print(key, dataFields[key].shape)
33             geolocation=file['HDFEOS']['SWATHS']['ColumnAmountNO2']['Geolocation Fields']
34         elif 'SO2' in FILE_NAME:
35             print('This is an OMI SO2 file. Saving... ')
36             SDS=dict({'1','ColumnAmountSO2_PBL'},(2,'ColumnAmountO3'),(3,'QualityFlags_PBL'))
37             dataFields=file['HDFEOS']['SWATHS']['OMI Total Column Amount SO2']['Data Fields']
38             geolocation=file['HDFEOS']['SWATHS']['OMI Total Column Amount SO2']['Geolocation Fields']
39         else:
40             print('The file named :',FILE_NAME, ' is not a valid OMI file. \n')
41             #if the program is unable to determine that it is an OMI SO2 or NO2 file, then it will skip to the next file
42             continue
43
44         #get Lat and Lon info as vectors
45         lat=geolocation['Latitude'][:].ravel()
46         lon=geolocation['Longitude'][:].ravel()
47
48         #get scan time and turn it into a vector
49         scan_time=geolocation['Time'][:].ravel()
50
51         #creates arrays the same size as scan time to receive time attributes
52         year=np.zeros(scan_time.shape[0])
53         month=np.zeros(scan_time.shape[0])
54         day=np.zeros(scan_time.shape[0])
55         hour=np.zeros(scan_time.shape[0])
56         min=np.zeros(scan_time.shape[0])
57         sec=np.zeros(scan_time.shape[0])
```

Python 3.6.2 [Anaconda custom (64-bit)] (default, Sep 19 2017, 08:03:39) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 6.1.0 -- An enhanced Interactive Python.
In [1]: runfile('C:/Users/mfcook/Documents/ARSET/Trainings - Current/Advanced Tools Webinar/Python codes/read_omi_no2_so2_and_dump_ascii.py', wdir='C:/Users/mfcook/Documents/ARSET/Trainings - Current/Advanced Tools Webinar/Python codes')
Would you like to process
OMI-Aura_L2-OMNO2_2008m0720t2016-o21357_v003-2016m0820t102252.he5
(Y/N)
This is an OMI NO2 file. Saving..
All files have been saved successfully.
Would you like to process
OMI-Aura_L2-OMNO2_2015m0615t2010-o58069_v003-2016m0821t1121351.he5
(Y/N)
This is an OMI NO2 file. Saving..
All files have been saved successfully.
Would you like to process
OMI-Aura_L2-OMSO2_2015m0616t0921-o58077_v003-2016m0621t111522.he5
(Y/N)
This is an OMI SO2 file. Saving..
All files have been saved successfully.
In [2]: |



Output

```
OMI-Aura_L2-OMNO2_2008m0720t2016-o21357_v003-2016m0820t102252 - Notepad
File Edit Format View Help
Year,Month,Day,Hour,Minute,Second,Latitude,Longitude,ColumnAmountNO2,ColumnAmountNO2Std,vcdqualityFlags
2008.0.7.0.20.0.20.0.38.0.37.0,-74.78585052490234,-121.10774993896484,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-74.95536041259766,-116.02945709228516,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-75.00605010986328,-111.8778076171875,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-74.98909759521484,-108.39921569824219,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-74.93197631835938,-105.4294204711914,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-74.85054779052734,-102.85506439208984,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-74.75444030761719,-100.59456634521484,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-74.649658203125,-98.58740234375,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-74.54005432128906,-96.78759002685347,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-74.42811584472656,-95.15950775146484,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-74.3154525756836,-93.67511749267578,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-74.20308685302734,-92.31201171875,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-74.09165954589844,-91.05208587646484,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-73.98152923583984,-89.88053131103516,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-73.87289428710938,-88.78511810302734,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-73.76580810546875,-87.75564575195312,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-73.66024017333984,-86.78353118896484,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-73.55611419677734,-85.86150360107422,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-73.45328521728516,-84.9833740234375,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-73.351604827632734,-84.14366128479797,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-73.25088500976562,-83.33782958984375,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-73.15093994140625,-82.56175994873047,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-73.05155944824219,-81.81185150146484,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-72.9525375366211,-81.08490753173828,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-72.8536605834961,-80.3780517578125,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-72.75472259521484,-79.6886978149414,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-72.65550231933594,-79.01445770263672,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-72.5557861328125,-78.35314178466797,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-72.45532989501953,-77.70271301269531,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-72.35382612730469,-77.06124877929294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-72.25133314404297,-76.42691802978316,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-72.14730072021484,-75.79795837402344,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-72.04157257080078,-75.17264556884766,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-71.93387603759766,-74.54930114746094,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-71.82392883300781,-73.92621612548828,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-71.7114028930664,-73.30171203613281,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-71.59595489501953,-72.6740493774414,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-71.47721099853516,-72.04144287109375,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-71.354736328125,-71.40204620361328,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-71.22806549072266,-70.75390625,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-71.09864154052734,-70.094970703125,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-70.95985412597656,-69.42302703857422,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-70.81696319580078,-68.73568725585938,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-70.66714477539062,-68.0303726196289,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-70.50940704345703,-67.30422973632812,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-70.34258270263672,-66.55410766601562,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-70.16527557373047,-65.7764892578125,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-69.97581481933594,-64.96739959716797,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-69.77216339111328,-64.12232208251953,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-69.55181884765625,-63.23603820800781,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-69.31162365478516,-62.30249786376953,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-69.0478599189453,-61.31453704833984,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-68.75537109375,-60.26359176635742,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-68.4277572631836,-59.13920211791992,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-68.05646514892578,-57.928348541259766,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-67.62984466552734,-56.61440658569336,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-67.13152313232422,-55.175418853759766,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-66.53754425048828,-53.5811882019043,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-65.8108139038086,-51.78777313232422,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.37.0,-64.88919830322266,-49.7259521484375,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.39.0,-74.67247009277344,-120.9896244024375,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.39.0,-74.83917236328125,-115.93070806873353,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.39.0,-74.88825225830078,-111.8367080684766,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.39.0,-74.87043762207031,-108.38494873046875,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.39.0,-74.81291961669922,-105.43817138671875,-1.2676506002282294e+30,3.0
2008.0.7.0.20.0.20.0.38.0.39.0,-74.73140716552734,-102.88363647460938,-1.2676506002282294e+30,3.0
```

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 SDS to be written as output

NOTE: This code will only work when all the variables listed are the same dimension. Use the “list SDS” code to view the variable dimensions

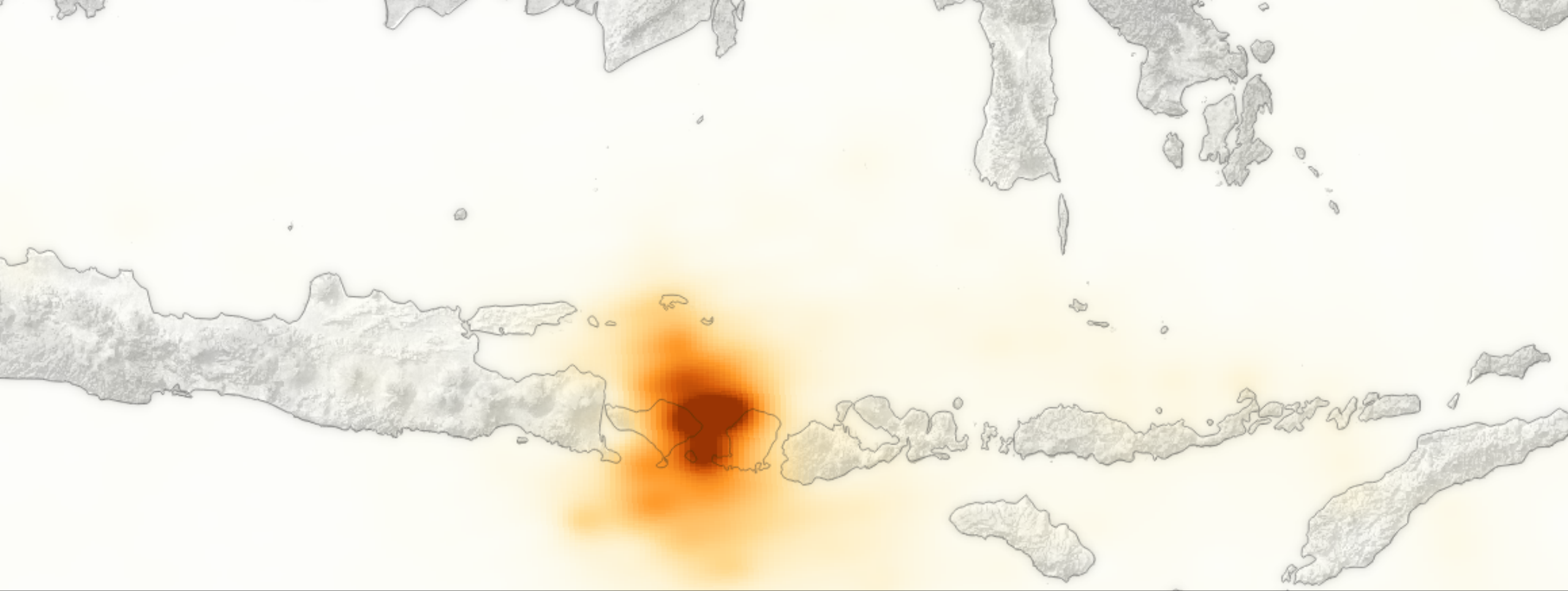
```
15 #Loops through all files listed in the text file
16 for FILE_NAME in fileList:
17     FILE_NAME=FILE_NAME.strip()
18     user_input=input('\nWould you like to process\n' + FILE_NAME + '\n\n(Y/N)')
19     if(user_input == 'N' or user_input == 'n'):
20         print('Skipping...')
21         continue
22     else:
23         file = h5py.File(FILE_NAME, 'r') # 'r' means that hdf5 file is open in read-only mode
24         #checks if the file contains NO2 or SO2 data, and reacts accordingly
25         if 'NO2' in FILE_NAME:
26             print('This is an OMI NO2 file. Saving... ')
27             #utilizes a python dictionary to determine the variable specified by user input
28             SDS=dict([(1,'ColumnAmountNO2'),(2,'ColumnAmountNO2Std'),(3,'VcdQualityFlags')])
29             #this is how you access the data tree in an hdf5 file
30             dataFields=file['HDFEOS']['SWATHS']['ColumnAmountNO2']['Data Fields']
31             #for key in dataFields:
32             #Y     print(key, dataFields[key].shape)
33             geolocation=file['HDFEOS']['SWATHS']['ColumnAmountNO2']['Geolocation Fields']
34         elif 'SO2' in FILE_NAME:
35             print('This is an OMI SO2 file. Saving... ')
36             SDS=dict([(1,'ColumnAmountSO2_PBL'),(2,'ColumnAmountO3'),(3,'QualityFlags_PBL')])
37             dataFields=file['HDFEOS']['SWATHS']['OMI Total Column Amount SO2']['Data Fields']
38             geolocation=file['HDFEOS']['SWATHS']['OMI Total Column Amount SO2']['Geolocation Fields']
39         else:
40             print('The file named :',FILE_NAME, ' is not a valid OMI file. \n')
41             #if the program is unable to determine that it is an OMI SO2 or NO2 file, then it will skip to the next file
42             continue
43
```



Applications

- This is a sample code to read and extract OMI Level 2 NO₂ and SO₂ data
- The code can be modified to extract varying SDSs into a single .csv file
- The code be easily modified to extract data over a certain region
- The output file can be opened in excel, or any other data analysis tool





Questions & Answers