TEMPO Product Training Overview

Amy K. Huff[1] and Barron H. Henderson[2]

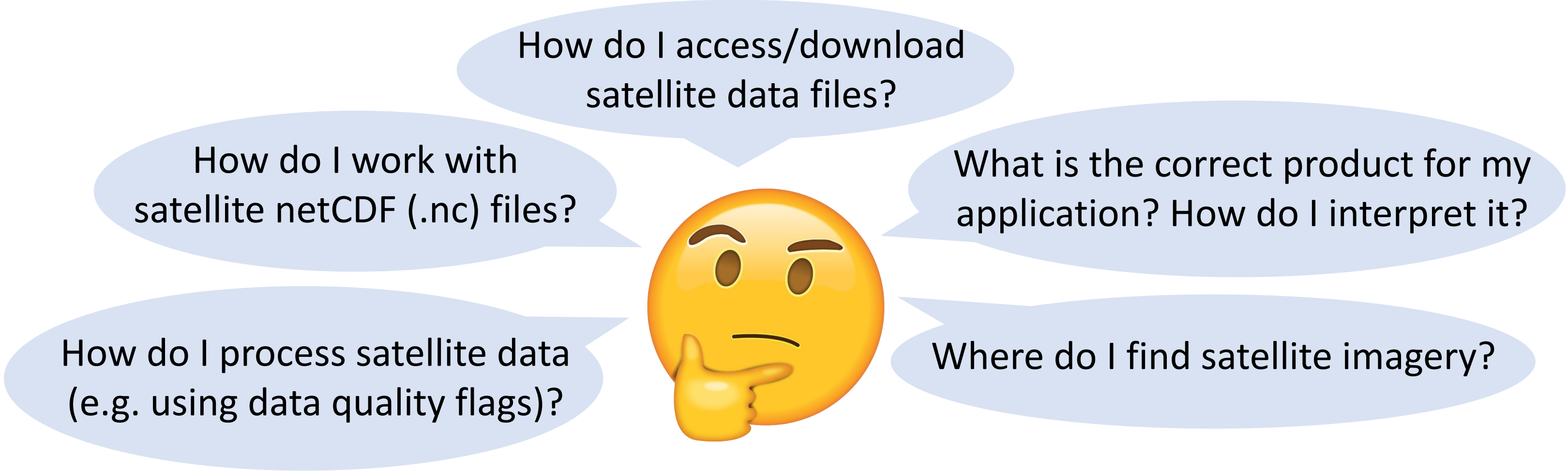
[1]IMSG at NOAA/NESDIS/STAR
[2]U.S. EPA

Session III: TEMPO
GeoXO ACX Science Team Meeting
May 7, 2024

NOAA
National Satellite and Information Service
Center for Satellite Applications and Research

Disclaimer: The scientific results and conclusions, as well as any views or opinions expressed herein, are those of the author(s) and do not necessarily reflect those of NOAA or the Department of Commerce.

STAR Atmospheric Composition Training Program



How do I access/download satellite data files?

How do I work with satellite netCDF (.nc) files?

What is the correct product for my application? How do I interpret it?

How do I process satellite data (e.g. using data quality flags)?

Where do I find satellite imagery?

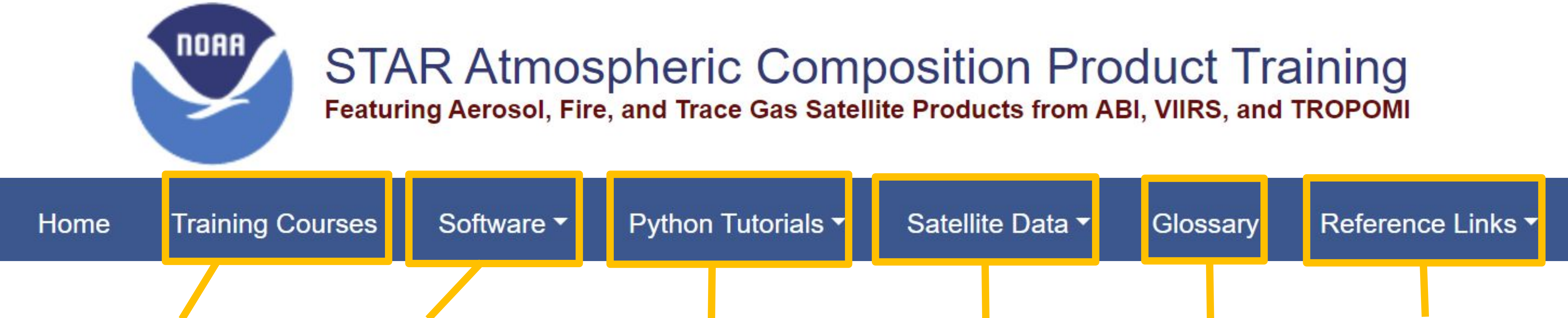
- Hands-on training program for end users; began in 2020
- Breaking down barriers preventing wider user of STAR products
- Targeted sessions for specialized user groups (e.g., air quality forecasters) and more general sessions at scientific meetings
- Focus is on **proper workflow** for satellite data using **Python**:
  - Search for/download satellite data files from online archives
  - Open/read netCDF satellite data files
  - Process/analyze/visualize satellite data

STAR Training Program Objectives

1. Increase access to STAR's aerosol, fire & trace gas satellite data
2. Promote proper use of the satellite products in operations & research

STAR Training Website: Resources for End Users

https://www.star.nesdis.noaa.gov/atmospheric-composition-training/



STAR Atmospheric Composition Product Training

Featuring Aerosol, Fire, and Trace Gas Satellite Products from ABI, VIIRS, and TROPOMI

Training Courses: Agendas & links to hands-on tutorials for specific training courses

Software: Instructions & links to install Anaconda & Python packages, set-up Jupyter Notebook

Python Tutorials: Examples of annotated Python code used in training tutorials

(additional tutorials coming soon!)

Satellite Data: Everything users need to work with satellite data files, including links to online data archives

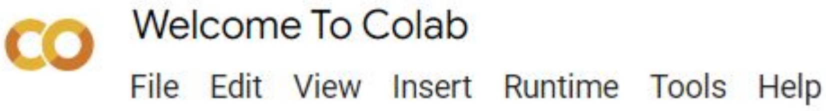
Glossary: Extensive glossary of satellite sensor and data terminology

Reference Links: Links to info on satellites, imagery, and Python resources

Thanks to Donovan Kelly, Jr. (Axiom Consulting) & Lori Brown (Axiom Consulting) for design of the website

TEMPO Product Training at the GeoXO ACX STM

- 2 opportunities to work hands-on with TEMPO data using Python
  - Tomorrow at 2:00-3:30 pm: **work flow for TEMPO NO_2 Level 2 granules & experimental AOD granules files** (parallel with the Poster Session)
  - Thursday at 1:00-2:00 pm: **TEMPO via the Python interface to the Remote Sensing Information Gateway (RSIG)** (parallel with end of lunch & beginning of Special Topics session)
- Python tutorials will be run on Google Colaboratory (Colab)
  - Google-hosted Jupyter Notebook service
  - No setup to use! **All you need is a Google (gmail) account!**
  - Free access to computing resources! Very fast & powerful!


Welcome To Colab

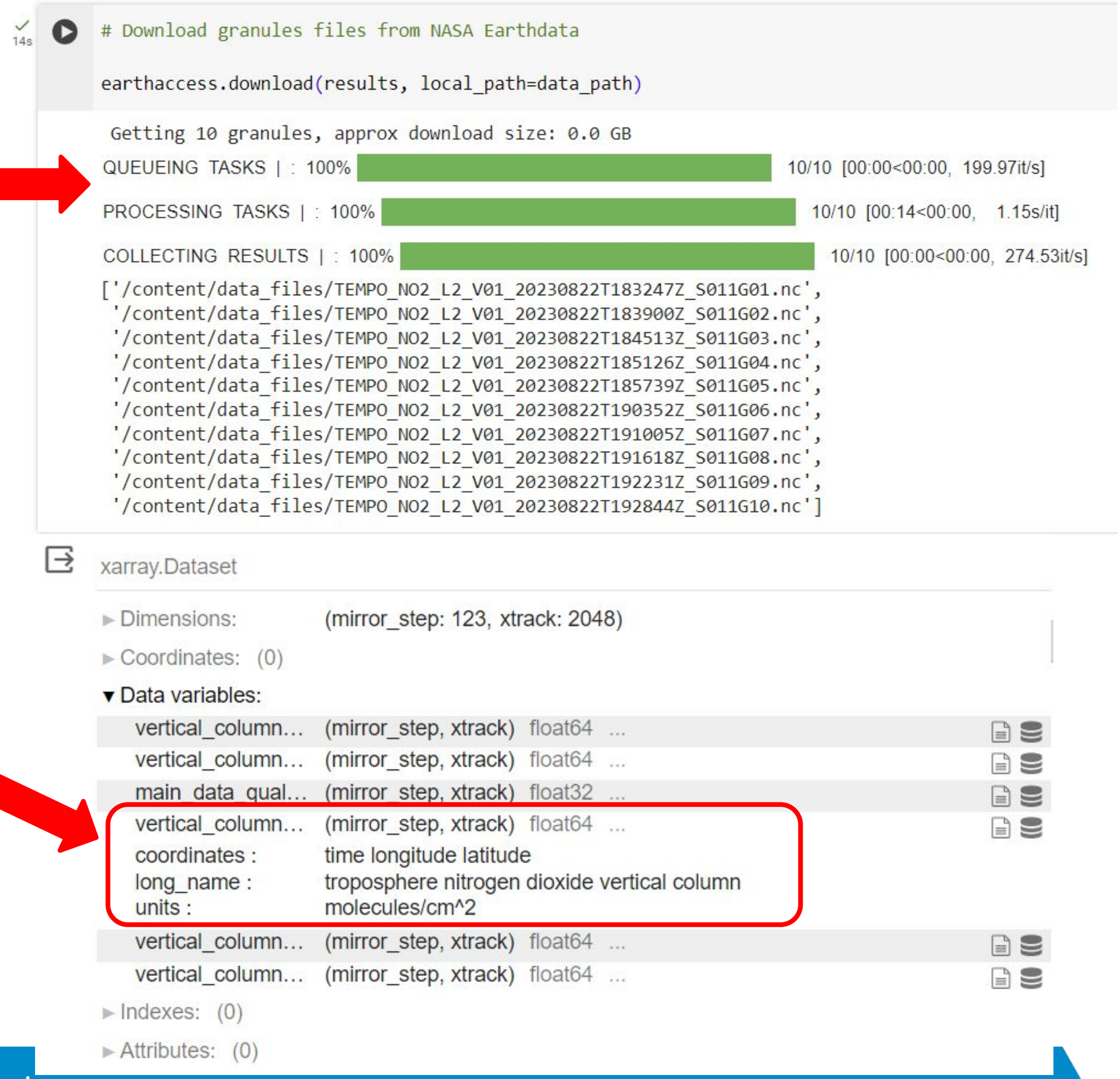
- Prior experience with Python or computer programming is NOT required
  - Come and see how easy it is to use Python!
  - Get your hands on actual TEMPO data!
  - More advanced Python programmers will find plenty to interest them!

Wednesday’s Training: Working with TEMPO Granules

- Download one full TEMPO scan (10 files) of:
  - Level 2 NO_2 granules from **NASA Earthdata**
  - Experimental aerosol granules from my STAR FTP webpage

If you don’t already have an Earthdata account, search for “Earthdata Login” and register! It’s free!

- Open & understand the contents of the TEMPO granule files
  - Find data variables; understand their valid ranges, units, data types
  - Tricks & tips for reading in data arrays
  - See the difference b/w an operational file (NO_2) & an experimental file (AOD)



```
# Download granules files from NASA Earthdata

earthaccess.download(results, local_path=data_path)
```

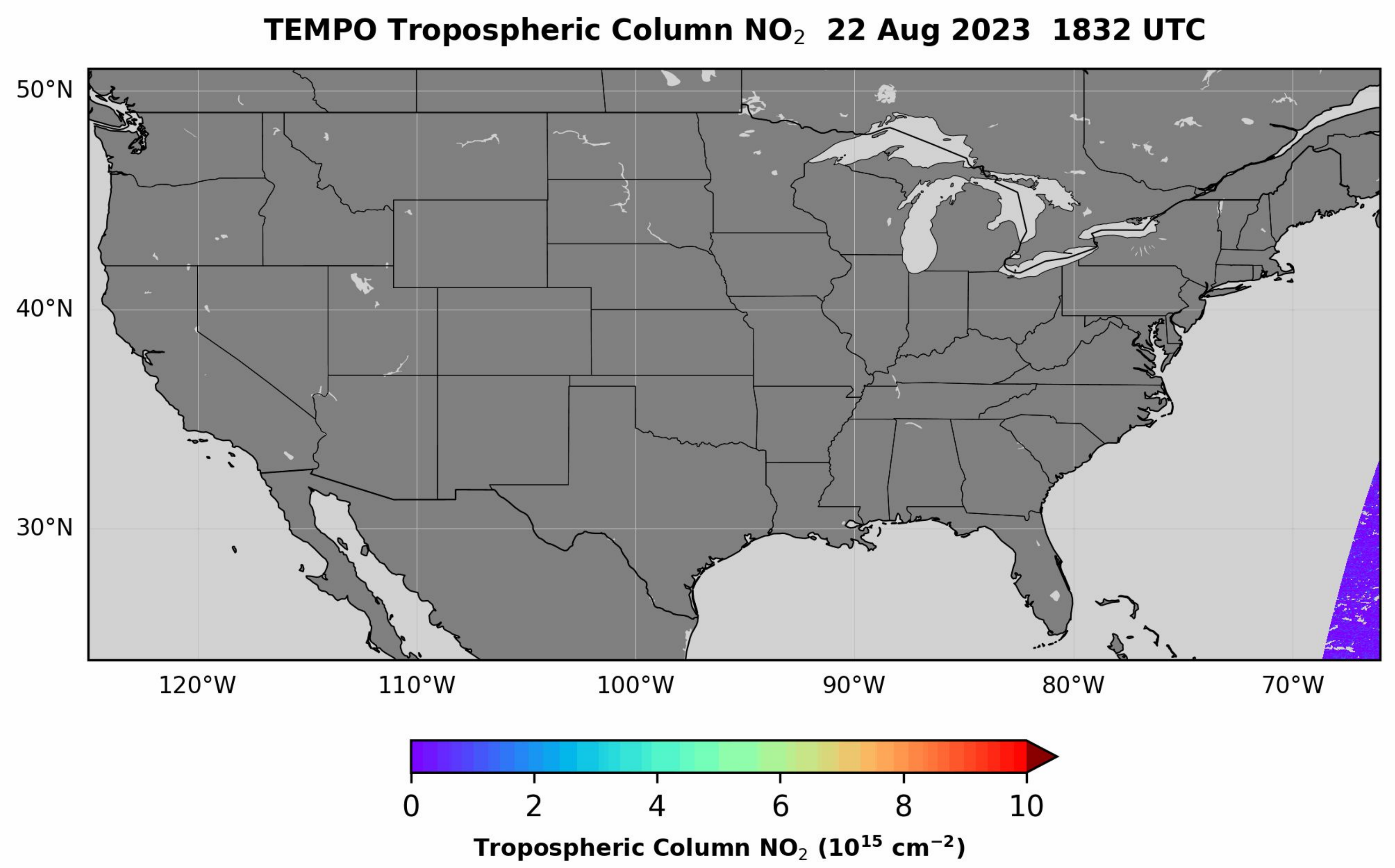
```
 Getting 10 granules, approx download size: 0.0 GB
QUEUEING TASKS | : 100%  10/10 [00:00<00:00, 199.97it/s]
PROCESSING TASKS | : 100%  10/10 [00:14<00:00,  1.15s/it]
COLLECTING RESULTS | : 100%  10/10 [00:00<00:00, 274.53it/s]
['/content/data_files/TEMPO_NO2_L2_V01_20230822T183247Z_S011G01.nc',
 '/content/data_files/TEMPO_NO2_L2_V01_20230822T183900Z_S011G02.nc',
 '/content/data_files/TEMPO_NO2_L2_V01_20230822T184513Z_S011G03.nc',
 '/content/data_files/TEMPO_NO2_L2_V01_20230822T185126Z_S011G04.nc',
 '/content/data_files/TEMPO_NO2_L2_V01_20230822T185739Z_S011G05.nc',
 '/content/data_files/TEMPO_NO2_L2_V01_20230822T190352Z_S011G06.nc',
 '/content/data_files/TEMPO_NO2_L2_V01_20230822T191005Z_S011G07.nc',
 '/content/data_files/TEMPO_NO2_L2_V01_20230822T191618Z_S011G08.nc',
 '/content/data_files/TEMPO_NO2_L2_V01_20230822T192231Z_S011G09.nc',
 '/content/data_files/TEMPO_NO2_L2_V01_20230822T192844Z_S011G10.nc']
```

```
xarray.Dataset

▸ Dimensions:        (mirror_step: 123, xtrack: 2048)
▸ Coordinates:  (0)
▾ Data variables:
    vertical_column…   (mirror_step, xtrack)  float64  …
    vertical_column…   (mirror_step, xtrack)  float64  …
    main_data_qual…    (mirror_step, xtrack)  float32  …
    vertical_column…   (mirror_step, xtrack)  float64  …
    coordinates :      time longitude latitude
    long_name :        troposphere nitrogen dioxide vertical column
    units :            molecules/cm^2
    vertical_column…   (mirror_step, xtrack)  float64  …
    vertical_column…   (mirror_step, xtrack)  float64  …
▸ Indexes:  (0)
▸ Attributes:  (0)
```

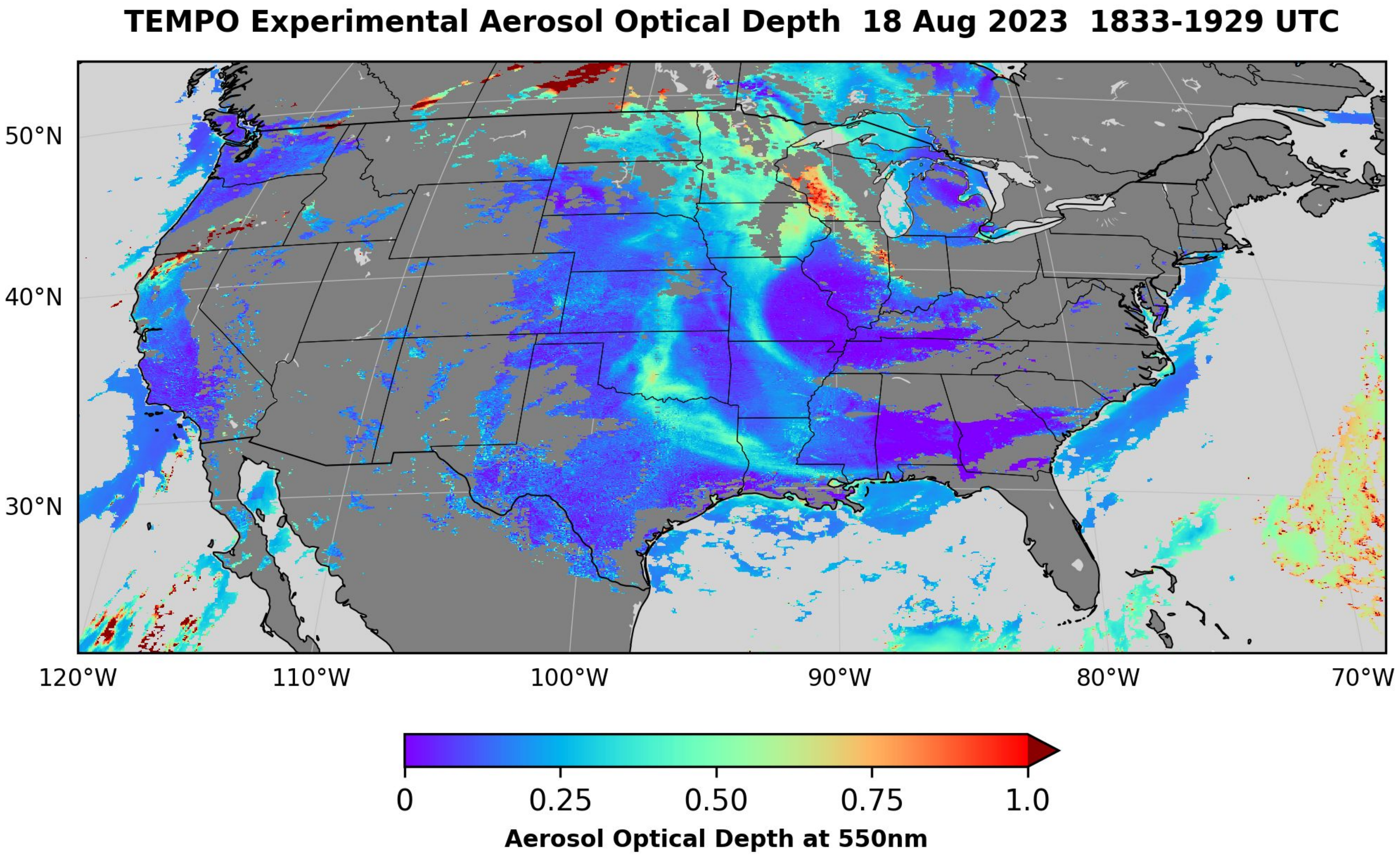
Wednesday’s Training: TEMPO NO_2 Granules

- Process tropospheric column NO_2 per TEMPO Science Team recommendations:
  - Set main_data_quality_flag = 0
  - Exclude pixels with effective_cloud_fraction > 0.3
  - Exclude pixels with solar_zenith_angle > 70
- Plot processed NO_2 from each granule on a separate map (Plate Carree equirectangular projection)
- Make a .gif animation of the individual NO_2 map plots


TEMPO Tropospheric Column NO₂ 22 Aug 2023 1832 UTC
50°N
40°N
30°N
120°W
110°W
100°W
90°W
80°W
70°W
0
2
4
6
8
10
Tropospheric Column NO₂ (10¹⁵ cm⁻²)

Wednesday's Training: TEMPO Experimental AOD Granules

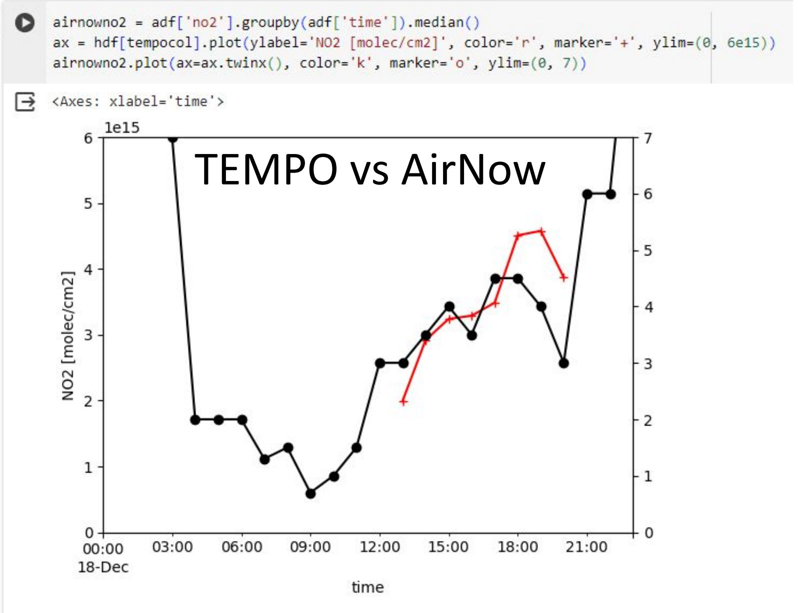
- Set up the geostationary map projection for the Intelsat-40e satellite
- Plot experimental AOD from each granule on one map (geostationary projection)
  - Thanks to Hai Zhang for providing the experimental TEMPO aerosol granules
  - Thanks to NASA and SAO for providing the TEMPO L1b data used in the aerosol retrieval algorithm


TEMPO Experimental Aerosol Optical Depth 18 Aug 2023 1833-1929 UTC
50°N
40°N
30°N
120°W
110°W
100°W
90°W
80°W
70°W
0
0.25
0.50
0.75
1.0
Aerosol Optical Depth at 550nm

Unofficial Experimental Data: Not for Public Release

Thursday’s Training: Remote Sensing Information Gateway (RSIG) Python and TEMPO

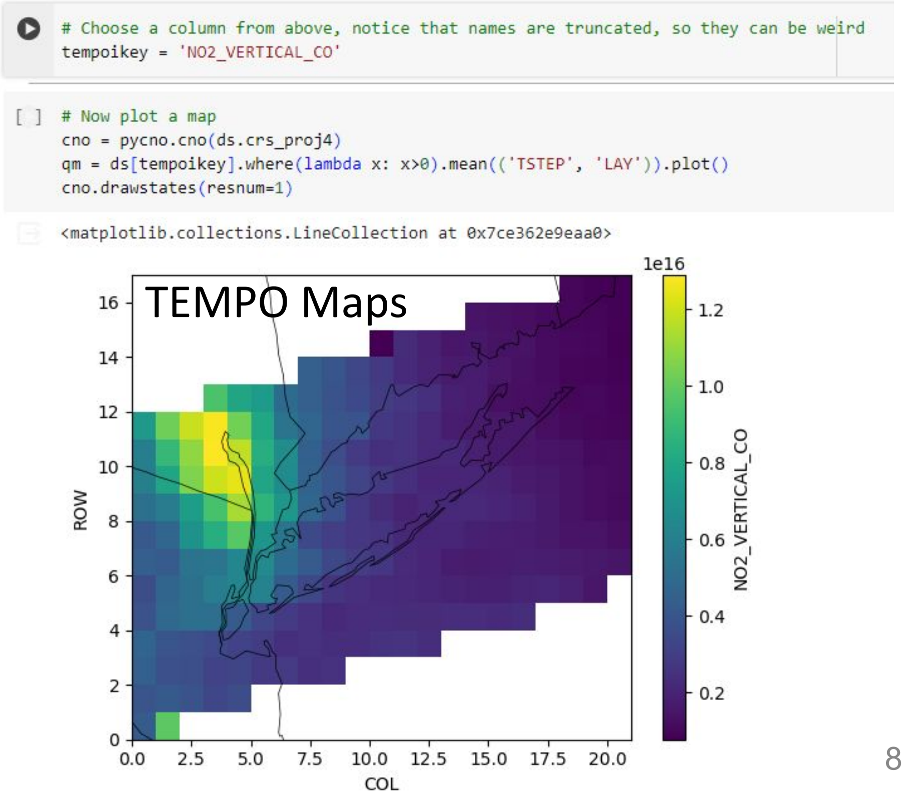
- RSIG makes satellite data easier
- RSIG connects satellite to AirNow, Pandora and many other observations
- Pyrsig uses python to script RSIG easily
- Training skills:
  - Compare TEMPO to AirNow, Pandora, TropOMI
  - Create a TEMPO NO2 map.
  - Create a TEMPO Surface NO2 estimate.
  - Adapt other tutorials.



```
airnowno2 = adf['no2'].groupby(adf['time']).median()
ax = hdf[tempocol].plot(ylabel='NO2 [molec/cm2]', color='r', marker='+', ylim=(0, 6e15))
airnowno2.plot(ax=ax.twinx(), color='k', marker='o', ylim=(0, 7))
```

```
<Axes: xlabel='time'>
```

TEMPO vs AirNow



```
# Choose a column from above, notice that names are truncated, so they can be weird
tempoikey = 'NO2_VERTICAL_CO'
```

```
# Now plot a map
cno = pycno.cno(ds.crs_proj4)
qm = ds[tempoikey].where(lambda x: x>0).mean(('TSTEP', 'LAY')).plot()
cno.drawstates(resnum=1)
```

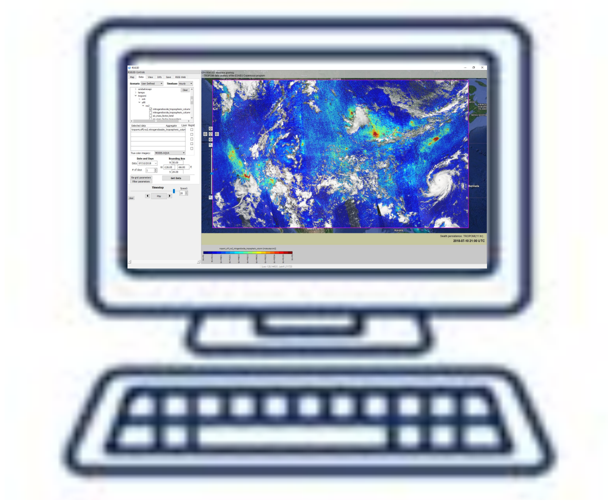
```
<matplotlib.collections.LineCollection at 0x7ce362e9eaa0>
```

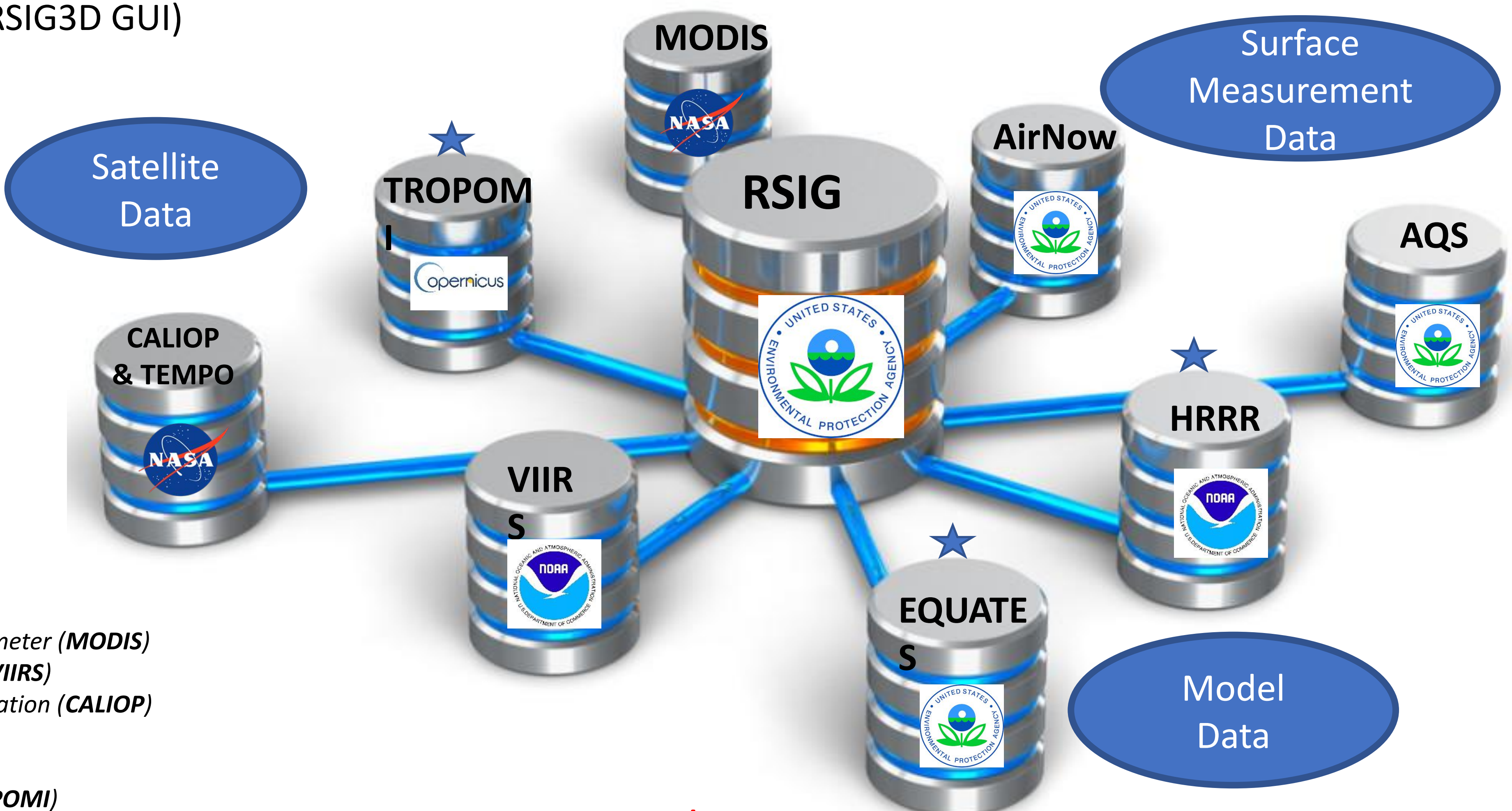
TEMPO Maps

What is the Remote Sensing Information Gateway?

- ***Free*** multi-platform, ***scriptable*** access to terabytes (TB) of air quality model, measurement, and satellite data from EPA, NOAA, NASA, ESA, etc.

- Multiple access methods
  - Graphical User Interface (RSIG3D GUI)
  - Python Interface (pyrsig)
  - Custom API shell scripts





Popular Air Quality Data Sets:

- *Moderate Resolution Imaging Spectroradiometer (**MODIS**)*
- *Visible Infrared Imaging Radiometer Suite (**VIIRS**)*
- *Cloud-Aerosol Lidar with Orthogonal Polarization (**CALIOP**)*
- *Air Quality System (**AQS**)*
- *EPA's Air QUAlity TimE Series (**EQUATES**)*
- *TROPOspheric Monitoring Instrument (**TROPOMI**)*
- *Tropospheric Emissions: Monitoring of POllution (**TEMPO**)*
- *High-Resolution Rapid Refresh (**HRRR**)*

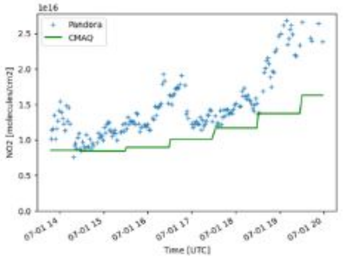
Training Focus : ***pyrsig/TEMPO***

Download Python source code: plot_elpaso.py

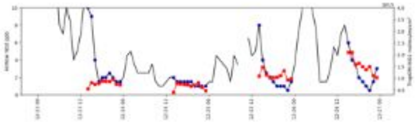
Download Jupyter notebook: plot_elpaso.ipynb

Python scripting examples in public gallery

https://barronh.github.io/pyrsig/

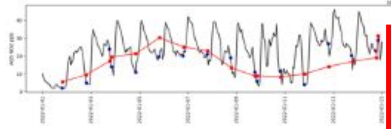


CMAQ vs Pandora

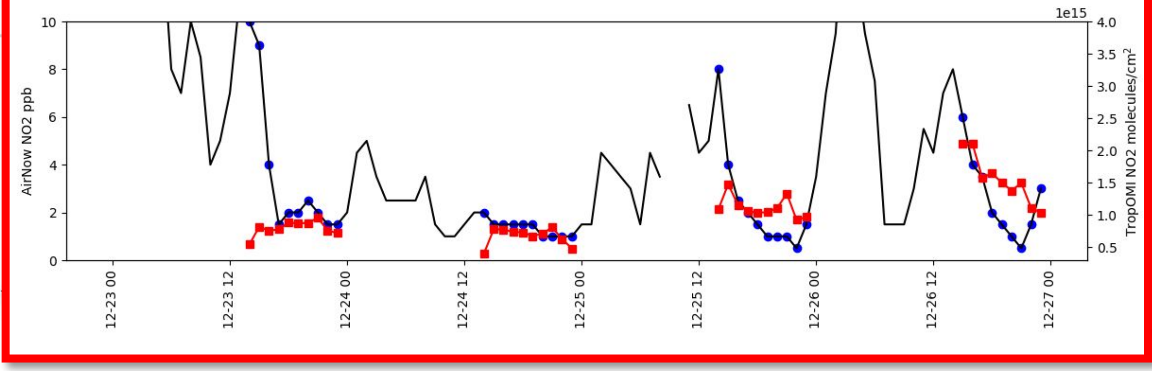


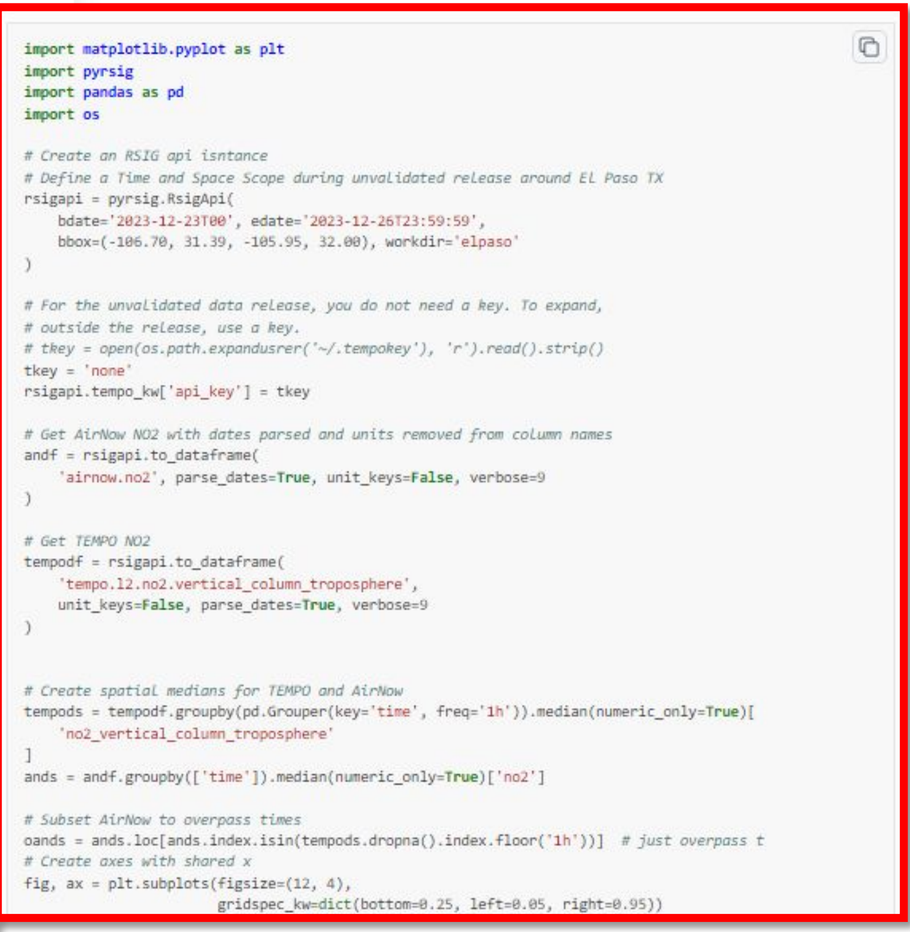
El Paso AirNow vs TEMPO

NYC VIIRS AOD vs TropOMI NO2



Phoenix AQS vs TropOMI





```
import matplotlib.pyplot as plt
import pyrsig
import pandas as pd
import os

# Create an RSIG api isntance
# Define a Time and Space Scope during unvalidated release around El Paso TX
rsigapi = pyrsig.RsigApi(
    bdate='2023-12-23T00', edate='2023-12-26T23:59:59',
    bbox=(-106.70, 31.39, -105.95, 32.00), workdir='elpaso'
)

# For the unvalidated data release, you do not need a key. To expand,
# outside the release, use a key.
# tkey = open(os.path.expanduser('~/.tempokey'), 'r').read().strip()
tkey = 'none'
rsigapi.tempo_kw['api_key'] = tkey

# Get AirNow NO2 with dates parsed and units removed from column names
andf = rsigapi.to_dataframe(
    'airnow.no2', parse_dates=True, unit_keys=False, verbose=9
)

# Get TEMPO NO2
tempodf = rsigapi.to_dataframe(
    'tempo.l2.no2.vertical_column_troposphere',
    unit_keys=False, parse_dates=True, verbose=9
)


# Create spatial medians for TEMPO and AirNow
tempods = tempodf.groupby(pd.Grouper(key='time', freq='1h')).median(numeric_only=True)[
    'no2_vertical_column_troposphere'
]
ands = andf.groupby(['time']).median(numeric_only=True)['no2']

# Subset AirNow to overpass times
oands = ands.loc[ands.index.isin(tempods.dropna().index.floor('1h'))]  # just overpass t
# Create axes with shared x
fig, ax = plt.subplots(figsize=(12, 4),
                       gridspec_kw=dict(bottom=0.25, left=0.05, right=0.95))
```

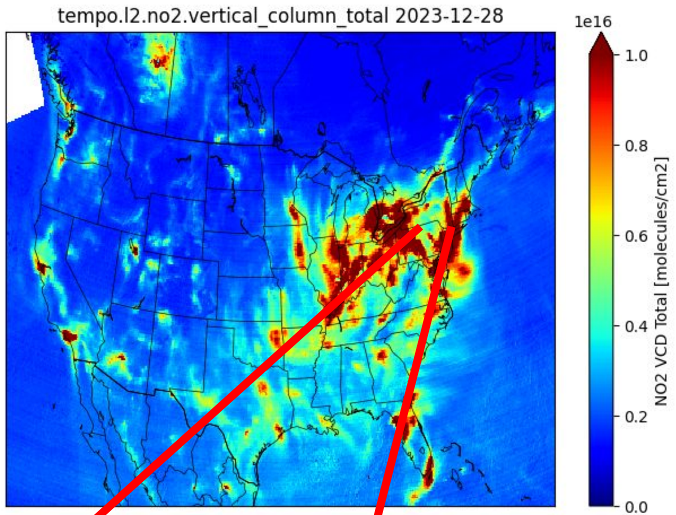
Pyrsig: Connecting TEMPO, TropOMI, and Pandora

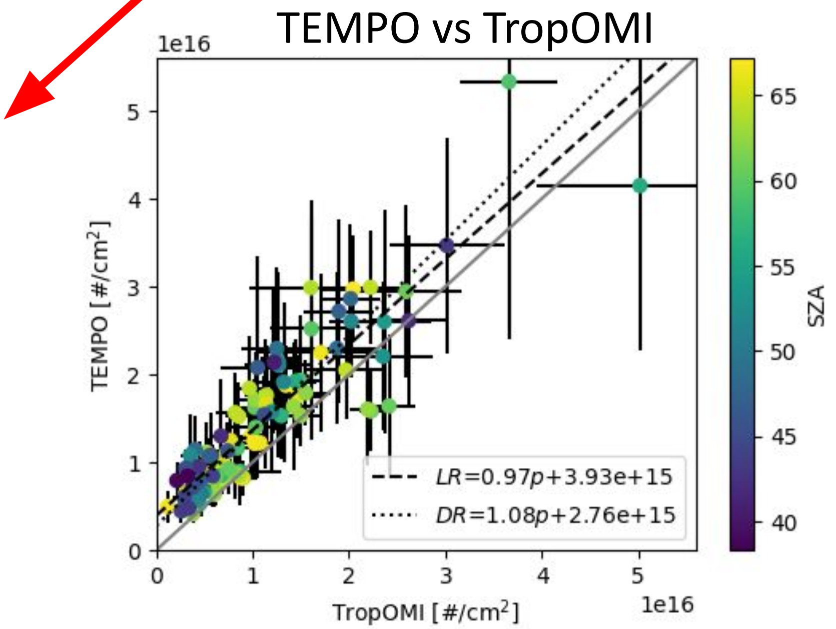
- pyrsig connects you to the same data ***and*** allows you to perform custom analyses.

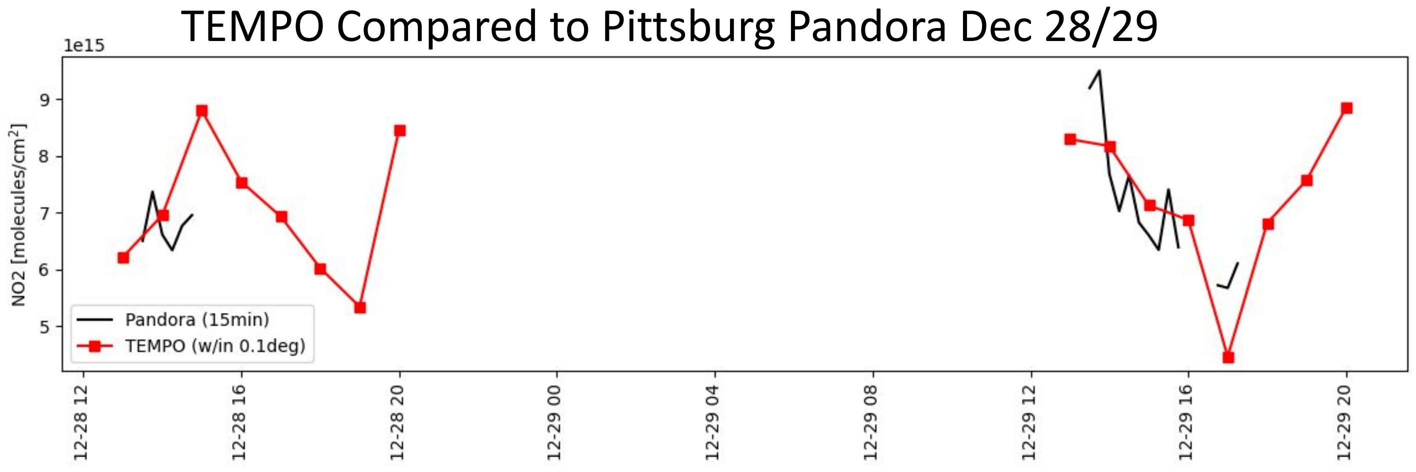
- Map uses TEMPO data that is already public (Dec 1-29)

- Scatterplot for long-term comparison between TEMPO and TropOMI.

- Time series on the bottom-right shows comparison to a Pandora ground-based remote sensor to evaluate TEMPO.


tempo.l2.no2.vertical_column_total 2023-12-28
1e16
1.0
0.8
0.6
0.4
0.2
0.0
NO2 VCD Total [molecules/cm2]


TEMPO vs TropOMI


TEMPO Compared to Pittsburg Pandora Dec 28/29

What You’ll Need for the TEMPO Training Sessions

- **Computer with access to the internet**
- **Google account (to use Google Colab)**
- Wednesday session only: NASA Earthdata account (to download TEMPO granules data files)
  - If you don’t have an account and don’t want one, it’s ok
  - Amy will provide a link to download the TEMPO NO_2 Level 2 data as a .zip file

In-person and virtual STM attendees are all welcome!