



TEMPO via the Python Interface to Remote Sensing Information Gateway

Presented: Barron H. Henderson

Special help: Jeff Willison, Fahim Sidi

Date: 2023-09-13

RSIG Team: Jim J. Szykman, Luke Valin, Todd Plessel, Matt Freeman

***Disclaimer:** The views expressed in this presentation are those of the authors and do not necessarily reflect the views or policies of the U.S. Environmental Protection Agency.*

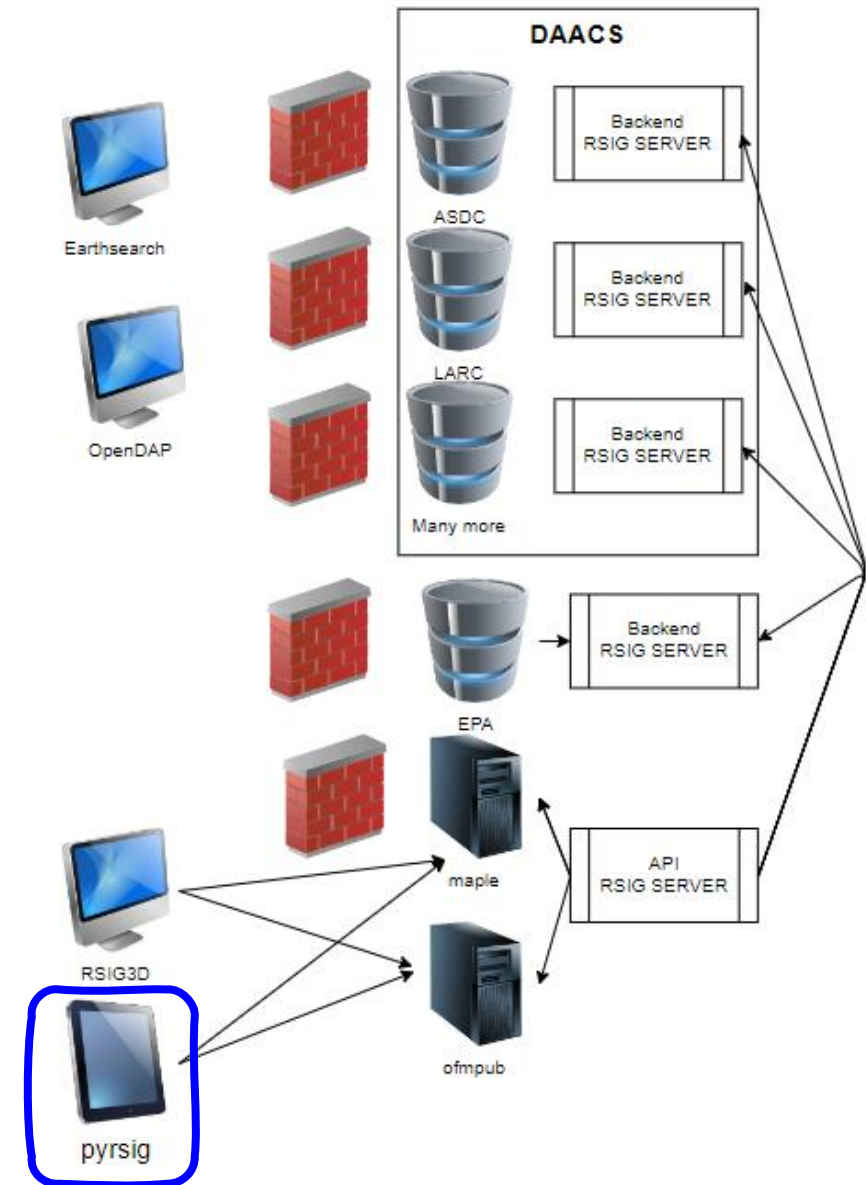


Overview

- Skills will be covered in the presentation
 - Setting up your python environment,
 - Using an example Jupyter notebook,
 - Retrieving TEMPO data using RSIG via pyrsig, and
 - Analyzing TEMPO data with example time series plots and maps
- Potential Applications
 - Assess how TEMPO agrees (or not) with available AQS monitors.
 - Compare TEMPO spatial gradients on stagnant days with emission data.
 - Compare TEMPO spatial gradients on all days with socioeconomic data.

RSIG is EPA's gateway to remote sensing data

- RSIG is a server and client system
- RSIG server is really a web of servers (EPA and NASA)
 - An EPA server provides a single front end application programming interface.
 - EPA and NASA servers provide services to the front end.
 - Provides satellite data, AQS data, EQUATES data, etc.
- RSIG3D is the client graphical user interface
 - Great for point and click analyses.
 - Not the focus today, see Jim Szykman, Luke Valin, Matt Freeman or Todd Plessel for more on that.
- And, **pyrsig** is a client for advanced analyses that leverages RSIG's easy access.
 - Great for custom analyses.
 - Runs in the cloud (AWS, Google Colab, or atmos)
 - Cloud apps work with workstation, phone, or tablet



pyrsig User's Guide

Python interface to RSIG Web API

RSIG server prepares the data!

The key value of *pyrsig* is to present RSIG data in pandas DataFrames and xarray Datasets. This makes it easy to do advanced analyses in a pythonic way. Example analyses are provided, but the sky is the limit.

Getting Started

The best way to get started is to install (see below) and then explore the examples gallery.

Installation

pyrsig is available through pypi.org, but is still in rapid development. You can get the latest release from pypi via the command below.

```
pip install pyrsig
```

Get data examples

Examples showing how to get data.

	STATION	pm25_corrected_hourly
0	131421	3.442696
1	119489	3.471192
2	15789	3.133492
3	108180	1.371232
4	8940	2.879763

Get DataFrame for PurpleAir PM25

	Timestamp	no2_tropospheric_column
0	2022-03-01T15:58:00-0000	6.214421e+14
1	2022-03-01T15:58:00-0000	7.000022e+14
2	2022-03-01T15:58:00-0000	2.961044e+14
3	2022-03-01T15:58:00-0000	5.383086e+14
4	2022-03-01T15:58:00-0000	7.535439e+14

Get DataFrame for TropOMI NO2

	Timestamp(UTC)	STATION(-)	ozone(ppb)
0	2022-05-01T00:00:00-0000	10030010	NaN
1	2022-05-01T00:00:00-0000	10499901	43.0
2	2022-05-01T00:00:00-0000	10510004	NaN
3	2022-05-01T00:00:00-0000	10550011	NaN
4	2022-05-01T00:00:00-0000	10730023	NaN

Get DataFrame for AQS ozone

xarray.Dataset		
Dimensions: (TSTEP: 24, VAR: 4, DATE-TIME: 2)		
Coordinates:		
TSTEP	(TSTEP)	datetime64[ns]
LAT	(LAT)	float64
ROW	(ROW)	int64
COL	(COL)	int64
Data variables:		
TRACED	(TSTEP VAR DATE-TIME)	float64
LONGITUDE	(TSTEP LAT ROW COL)	float64
LATITUDE	(TSTEP LAT ROW COL)	float64
COUNT	(TSTEP LAT ROW COL)	int64
NO2	(TSTEP LAT ROW COL)	float64
Attributes: (4)		
Attributes: (4)		

Get IOAPI formatted NetCDF TropOMI NO2

```
import pyrsig

rsigapi = pyrsig.RsigApi()
keys = rsigapi.keys()
print(len(keys), keys)
# 80 ('airnow.pm25', ... 'aqc.ozone', ...
# 'meteor.wind', ... 'pandora.ozone',
# 'tropomi.offl.no2.nitrogen dioxide_t1
# 'virsnoaa.frrad.A00550', ...)
keys = rsigapi.keys(offline=False) # slow
print(len(keys))
# 3875
```

Get List of Possible Coverages

xarray.Dataset		
Dimensions: (points: 89908)		
Coordinates: (0)		
Data variables:		
column	(points)	int32
row	(points)	int32
count	(points)	int32
longitude	(points)	float32
latitude	(points)	float32
no2	(points)	float32
time	(points)	datetime64[ns]

Get COARDS formatted NetCDF TropOMI NO2

Get data examples

Examples showing how to get data.

	STATION	pm25_corrected_hourly
0	131421	3.442696
1	119489	3.471192
2	15789	3.133492
3	108180	1.371232
4	8940	2.879763

Get DataFrame for PurpleAir PM25

	Timestamp	no2_tropospheric_column
0	2022-03-01T15:58:00-0000	6.214421e+14
1	2022-03-01T15:58:00-0000	7.000022e+14
2	2022-03-01T15:58:00-0000	2.961044e+14
3	2022-03-01T15:58:00-0000	5.383086e+14
4	2022-03-01T15:58:00-0000	7.535439e+14

Get DataFrame for TropOMI NO2

	Timestamp(UTC)	STATION(-)	ozone(ppb)
0	2022-05-01T00:00:00-0000	10090010	NaN
1	2022-05-01T00:00:00-0000	10499991	43.0
2	2022-05-01T00:00:00-0000	10510004	NaN
3	2022-05-01T00:00:00-0000	10550011	NaN
4	2022-05-01T00:00:00-0000	10730023	NaN

Get DataFrame for AQS ozone

xarray Dataset		
Dimensions:	(TSTEP: 24, VAR: 4, DATE-TIME: 2)	
Coordinates:		
TSTEP	(TSTEP)	seconds
LAT	(LAT)	degrees
ROW	(ROW)	int64
COL	(COL)	int64
Data variables:		
TRACED	(TSTEP VAR DATE-TIME)	float64
LONGITUDE	(TSTEP LAT ROW COL)	float64
LATITUDE	(TSTEP LAT ROW COL)	float64
COUNT	(TSTEP LAT ROW COL)	int64
NO2	(TSTEP LAT ROW COL)	float64
Indexes:	(4)	
Attributes:	(34)	

Get IOAPI formatted NetCDF TropOMI NO2

```
import pyrsig
```

```
rsigapi = pyrsig.RsigApi(bdate='2022-03-01')
```

```
df = rsigapi.to_dataframe('tropomi.offl.no2.nitrogen dioxide_tropospheric_column')
```

```
print(df.shape, *df.columns)
```

```
# (303444, 4) Timestamp(UTC) LONGITUDE(deg) LATITUDE(deg) nitrogen dioxide_tropospheric_column(molecules,
```

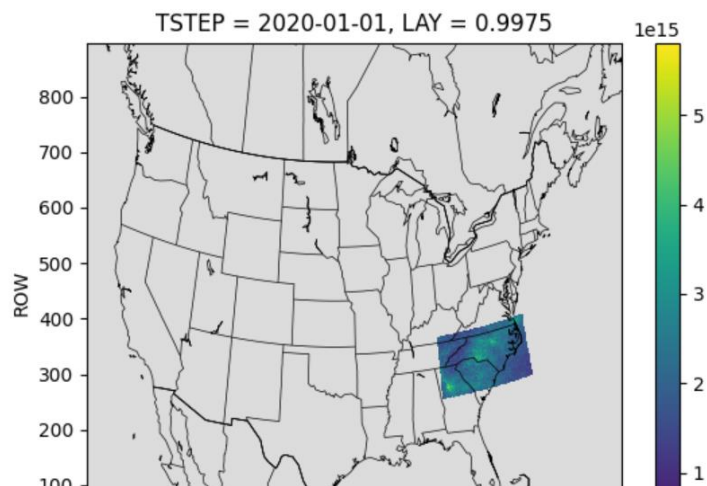
Oversample examples

Examples showing data oversampling of satellite data on



Oversample
CONUS at 4k

Previous



```
import matplotlib.pyplot as plt
import pyrsig
import pandas as pd
import xarray as xr
import pycno
import os

# Create a working directory
gdnam = '4US1'
bdate = '2021-01-01'
edate = '2021-01-15'
wdir = f'{gdnam}/{bdate[:4]}'

os.makedirs(gdnam, exist_ok=True)

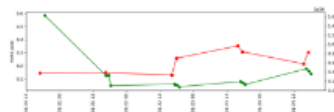
rsigapi = pyrsig.RsigApi(
    bdate=bdate, bbox=(-85, 33, -75, 37),
    encoding={'zlib': True, 'complevel': 1, '_FillValue': -9.999e36},
    workdir=wdir, grid_kw='4US1'
)

# Update to download daily averages instead of hourly
rsigapi.grid_kw['REGRID_AGGREGATE'] = 'daily'

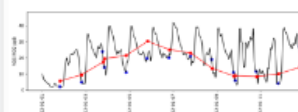
# Loop over days
dss = []
for bdate in pd.date_range('2020-01-01', '2020-01-15'):
    print(bdate)
    try:
        ds = rsigapi.to_ioapi(key='tropomi.offl.no2.nitrogendioxide_tropospheric_column', bdate=bdate)
        dss.append(ds)
    except Exception as e:
```


Timeseries examples

Examples showing timeseries analyses that illustrate the use of the pyrsig package

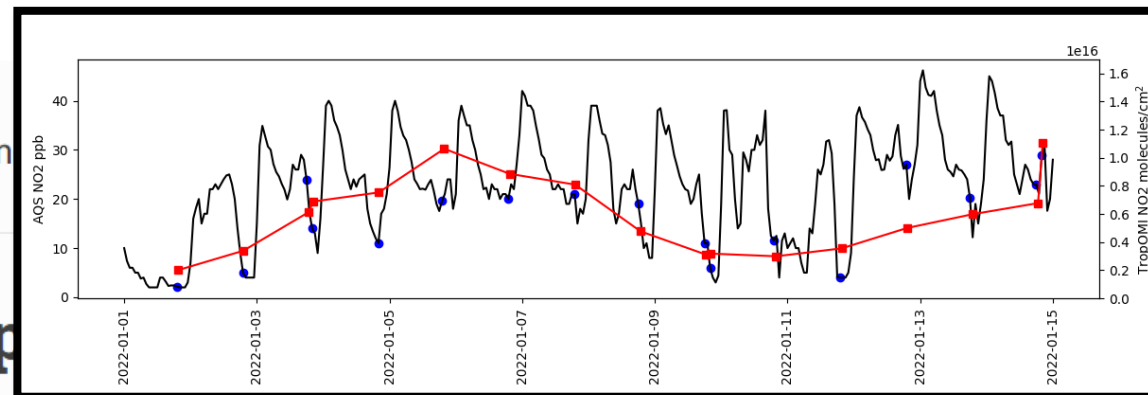


NYC VIIRS AOD vs TropOMI NO2



Phoenix AQS vs TropOMI

Previous



```
import matplotlib.pyplot as plt
import pyrsig

# Create an RSIG api instance
# Define a Time and Space Scope: here end of February around Phoenix
rsigapi = pyrsig.RsigApi(
    bdate='2022-01-01', edate='2022-01-15',
    bbox=(-112.3, 33.25, -111.85, 33.65)
)

# Get AQS NO2 with dates parsed and units removed from column names
aqsd = rsigapi.to_dataframe('aq.no2', parse_dates=True, unit_keys=False)

# Get TropOMI NO2
tomino2df = rsigapi.to_dataframe(
    'tropomi.offl.no2.nitrogendioxide_tropospheric_column',
    unit_keys=False, parse_dates=True
)

# Create spatial means for TropOMI and AQS
tomids = (
    tomino2df.groupby('time').median()['nitrogendioxide_tropospheric_column']
)
aqsd = aqsd.groupby(['time']).median()['no2']

# Subset AQS to overpass times
oaqsd = aqsd.loc[aqsd.index.isin(tomids.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))
ax.tick_params(axis='x', labelrotation = 90)
tax = ax.twinx()
```


How to run any example on Jupyter?

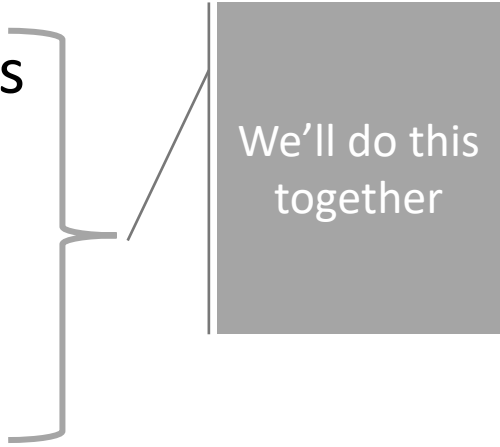
- Configure
 - Open a new notebook
 - Add “%pip install --user netcdf4 pyproj pycno pyrsig”
 - And run it (once on atmos; each time for Colab or Sagemaker)
- Go to <https://barronh.github.io/pyrsig/>
 - Navigate to example
 - Copy code from example
 - Paste it in your colab notebook
 - Click play button

If this is your first time running Jupyter on atmos:

Instructions are at “Jupyter via OnDemand” in <https://gaftp.epa.gov/Air/aqmg/bhenders/tutorials.html>

TEMPO Tutorial

- <https://gaftp.epa.gov/Air/aqmg/bhenders/tutorials.html>
 - Click on “TEMPO pyrsig training notebook (May 2023 meeting)”
 - The link takes you to a github gist.
- To run on Google Colab, click the “Open in Colab” badge
 - Click play on each cell
 - The first code cell will tell you to restart. Choose “runtime”, then “restart runtime”.
 - Then run the rest of the cells.
- To run on atmos, download the notebook and upload to atmos
 - Open JupyterHub on Atmos
 - Navigate to /home/bhenders/tutorials/pyrsig/tempo
 - Open SetupEnvironmetn.ipynb and run it
 - Open tempo_pyrsig.ipynb
 - Choose “File” “Save as” and save in your own space to run it



We'll do this together



Questions?

pyrsig: henderson.barron@epa.gov

RSIG: rsig@epa.gov

