PseudoNetCDF v3: A new interface and useful examples

Barron H. Henderson at the US EPA

Overview

PseudoNetCDF development began in 2006 at UNC. Many things have changed, and everything is easier. Simple core functions enable easy analyses or can be combined to make complex analyses.

How to:

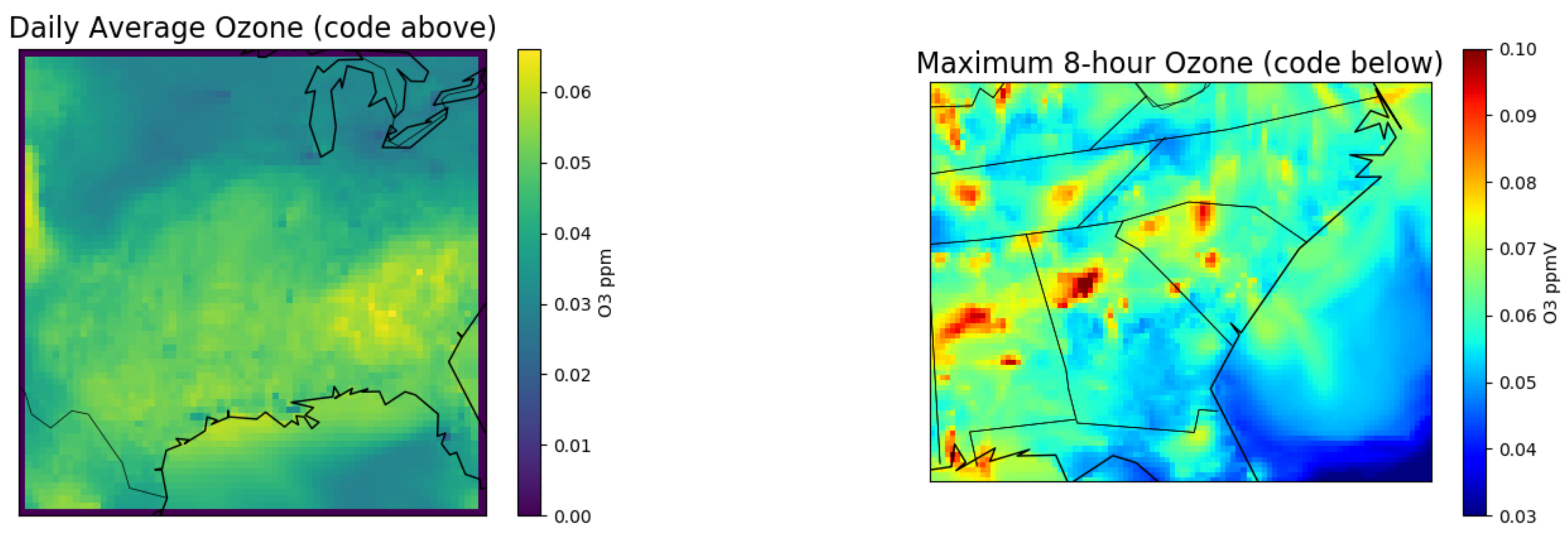
- pncopen : Returns CMAQ, CAMx, GEOS-Chem, MOZART4, or WRF file
- sliceDimensions : slices or windows file variables
- applyAlongDimensions : applies functions over dimensions
- ll2ij, time2t, interpSigma : convert between coordinates and indices
- plot : easy access plotting functionality via matplotlib
- more : mask, insertDimension, copyVariable, ...

Common Imports for All Examples

```
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import PseudoNetCDF as pnc
```

Plotting Maps Is Easy and Powerful

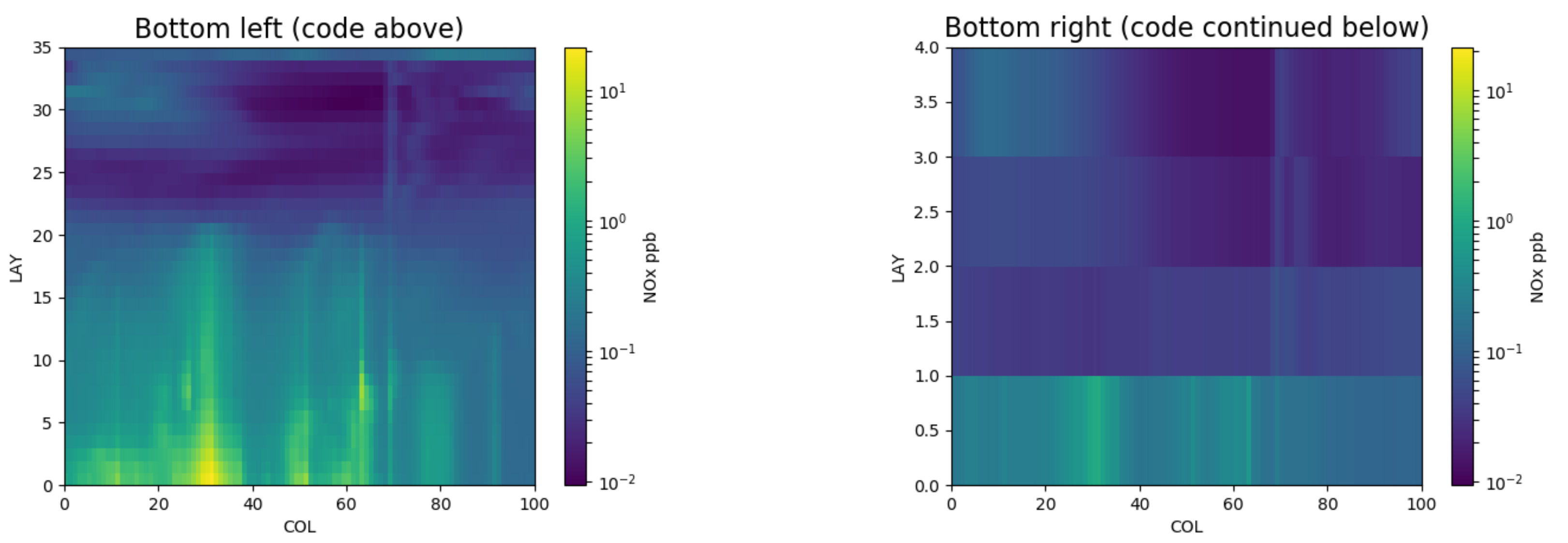
```
path = ’CAMx.v6.40.midwest.36.12.noMPI.20020603.avrg.grd01’
xf = pnc.pncopen(path, format=’uamiv’) # open CAMx uamiv file
ax = xf.plot(’O3’) # implicit time averaging
ax.set_title(’Daily␣Average␣Ozone␣(code␣above)’, fontsize=16)
ax.figure.savefig(’O3.png’)
```



```
def mda8(hourly):
    nhrs = 8
    a8 = np.convolve([1./nhrs]∗nhrs, hourly, mode=’full’)
    return a8[nhrs − 1:].reshape(−1, 24).max(1)

path = ’CCTM_ACONC_v52_cb6r3_intel17.0_SE52BENCH_20110701.nc’
qf = pnc.pncopen(path, format=’ioapi’).subsetVariables([’O3’])

wqf = qf.sliceDimensions(ROW=slice(1, −1), COL=slice(1, −1))
mqf = wqf.applyAlongDimensions(TSTEP=mda8)

opts = dict(cmap=’jet’, vmin=0.03, vmax=0.1)
ax = mqf.plot(’O3’, plot_kw=opts, map_kw=dict(states=True))
ax.set_title(’Maximum␣8−hour␣Ozone␣(code␣below)’, fontsize=16)
ax.figure.savefig(’O3_custom.png’)
```

Subsetting, Expressions, and Vertical Interp

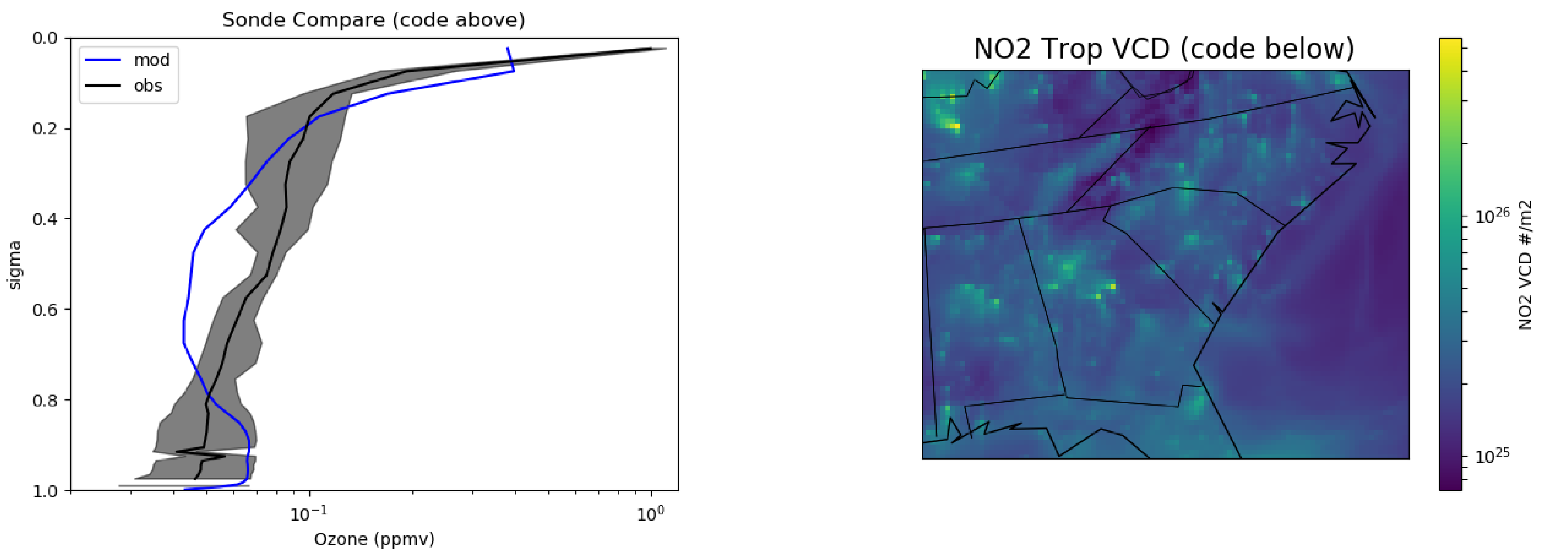
```
path = ’CCTM_CONC_v52_cb6r3_intel17.0_SE52BENCH_20110701.nc’
inpf = pnc.pncopen(path, format=’ioapi’)
# Use short aliases to get row 40 NO and NO2
rf = inpf.subset([’NO’, ’NO2’]).slice(ROW=40)
# Calculate NOx
ef = rf.eval(’NOx␣=␣(NO[:]␣+␣NO2[:])␣∗␣1000;␣NOx.units=”ppb”’)

opts = {’norm’: mpl.colors.LogNorm()}
ax = ef.plot(’NOx’, plottype=’COL−LAY’, plot_kw=opts)
ax.set_title(’Bottom␣left␣(code␣above)’, size=16)
```



```
# collapse to arbitrary 4 layer structure
zf = ef.interpSigma([1, .75, .5, .25, 0], vgtop=5000)
plt.close()
ax = zf.plot(’NOx’, plottype=’COL−LAY’, plot_kw=opts)
ax.set_title(’Bottom␣right␣(code␣continued␣below)’, size=16)
ax.figure.savefig(’NOx_newZ.png’)
```

Sonde Evaluation

```
from glob import glob
path = ’CCTM_CONC_v52_cb6r3_intel17.0_SE52BENCH_20110701.nc’
qf = pnc.pncopen(path, format=’ioapi’).subsetVariables([’O3’])

paths = glob(’data/hu???_2011_07_??_18.l100’)
sfs = [pnc.pncopen(path, format=’l100’) for path in paths]

lon, lat = sfs[0].variables[’longitude’], sfs[0].variables[’latitude’]
i, j = qf.ll2ij(lon, lat)

qsf = qf.apply(TSTEP=’mean’).slice(TSTEP=0∗j, ROW=j, COL=i, newdims=(’site’
 ,))

sqfs = [sf.avgSigma(qf.VGLVLS, vgtop=qf.VGTOP).mask(less=0) for sf in sfs]

so3 = np.ma.masked_less([sqf.variables[’O3’][0] for sqf in sqfs], 0)

z = (qf.VGLVLS[1:] + qf.VGLVLS[:−1]) / 2

ax = plt.gca(
    ylabel=’sigma’, xlabel=’Ozone␣(ppmv)’,
    xscale=’log’, xlim=(.02, 1.2), ylim=(1, 0)
)
ax.plot(qsf.variables[’O3’][0], z, label=’mod’, color=’blue’)
ax.plot(so3.mean(0), z, color=’k’, label=’obs’)
ax.fill_betweenx(y=z, x1=so3.min(0), x2=so3.max(0), color=’k’, alpha=.5)
ax.set_title(’Sonde␣Compare␣(code␣above)’)
plt.legend()
ax.figure.savefig(’sonde.png’)
```



Vertical Column Density (like Satellites)

```
path = ’CCTM_CONC_v52_cb6r3_intel17.0_SE52BENCH_20110701.nc’
cf = pnc.pncopen(path, format=’ioapi’).subsetVariables([’NO’, ’NO2’])

mpath = ’METCRO3D_110701.nc’
mf = pnc.pncopen(mpath, format=’ioapi’).subsetVariables([’DENS’, ’ZF’, ’ZH’
 ])
mf.copyVariable(cf.variables[’NO2’], key=’NO2’)

vf = mf.eval(”””DZ=ZF∗1;␣ZF[:,␣1:]␣=␣ZF[:,␣1:]␣−␣ZF[:,␣:−1];
NO2MOL␣=␣NO2␣∗␣DENS␣/␣0.287␣∗␣2␣∗␣DZ␣∗␣6.022e23
”””)
# approximate tropospheric top layers 0 to 29 (fortran 1 to 30)
vcdf = vf.sliceDimensions(LAY=slice(0, 30)).applyAlongDimensions(LAY=’sum’)
# plotting not shown, figure above
```

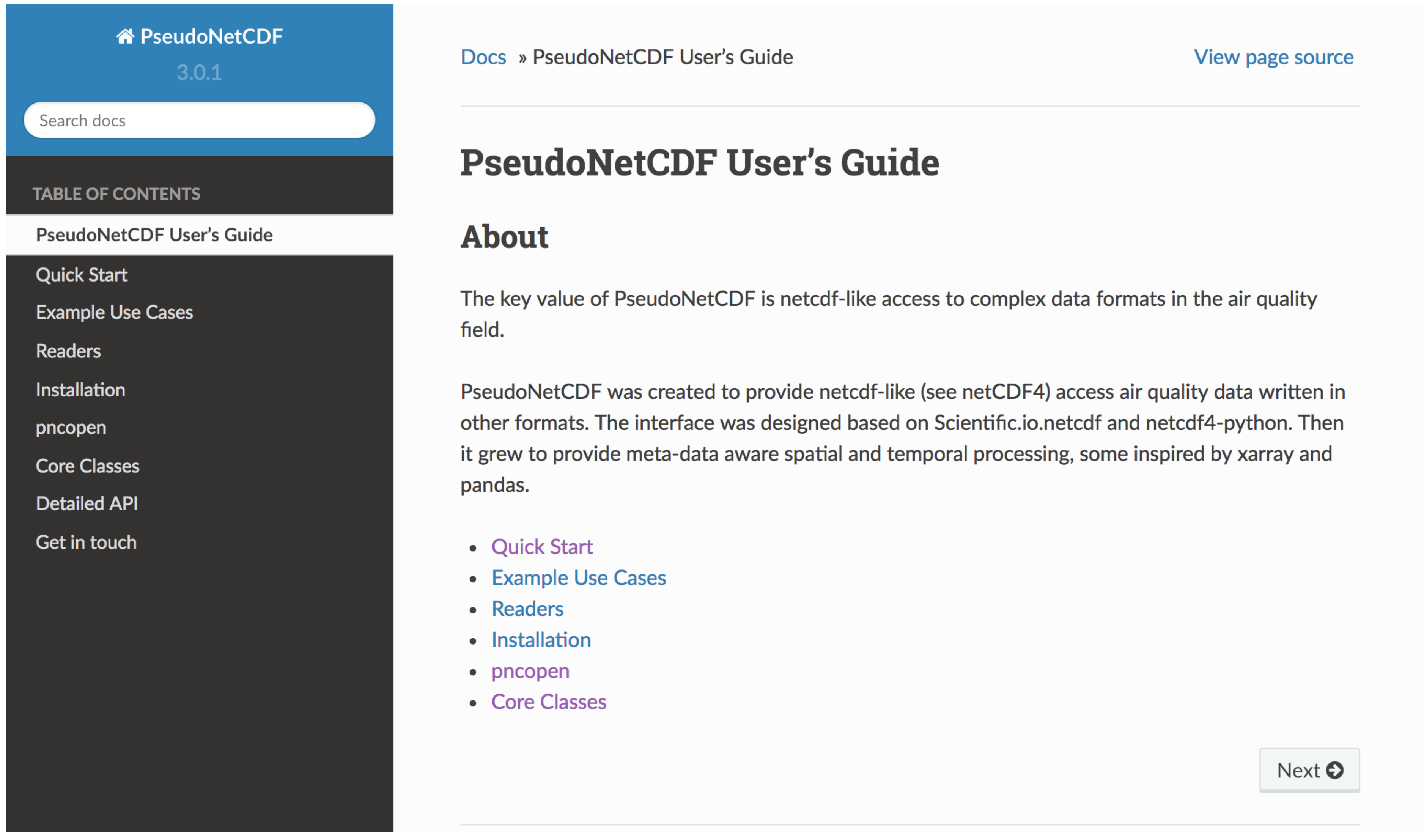
More Information

On Your Computer

- pydoc PseudoNetCDF
- python ‘help‘ function

Online (below)

- github.com/barronh/pseudonetcdf
- pseudonetcdf.readthedocs.io



PseudoNetCDF User’s Guide

About

The key value of PseudoNetCDF is netcdf-like access to complex data formats in the air quality field.

PseudoNetCDF was created to provide netcdf-like (see netCDF4) access air quality data written in other formats. The interface was designed based on Scientific.io.netcdf and netcdf4-python. Then it grew to provide meta-data aware spatial and temporal processing, some inspired by xarray and pandas.

- Quick Start
- Example Use Cases
- Readers
- Installation
- pncopen
- Core Classes

See also Byeong-Uk Kim’s poster and Barry Baker’s MONET