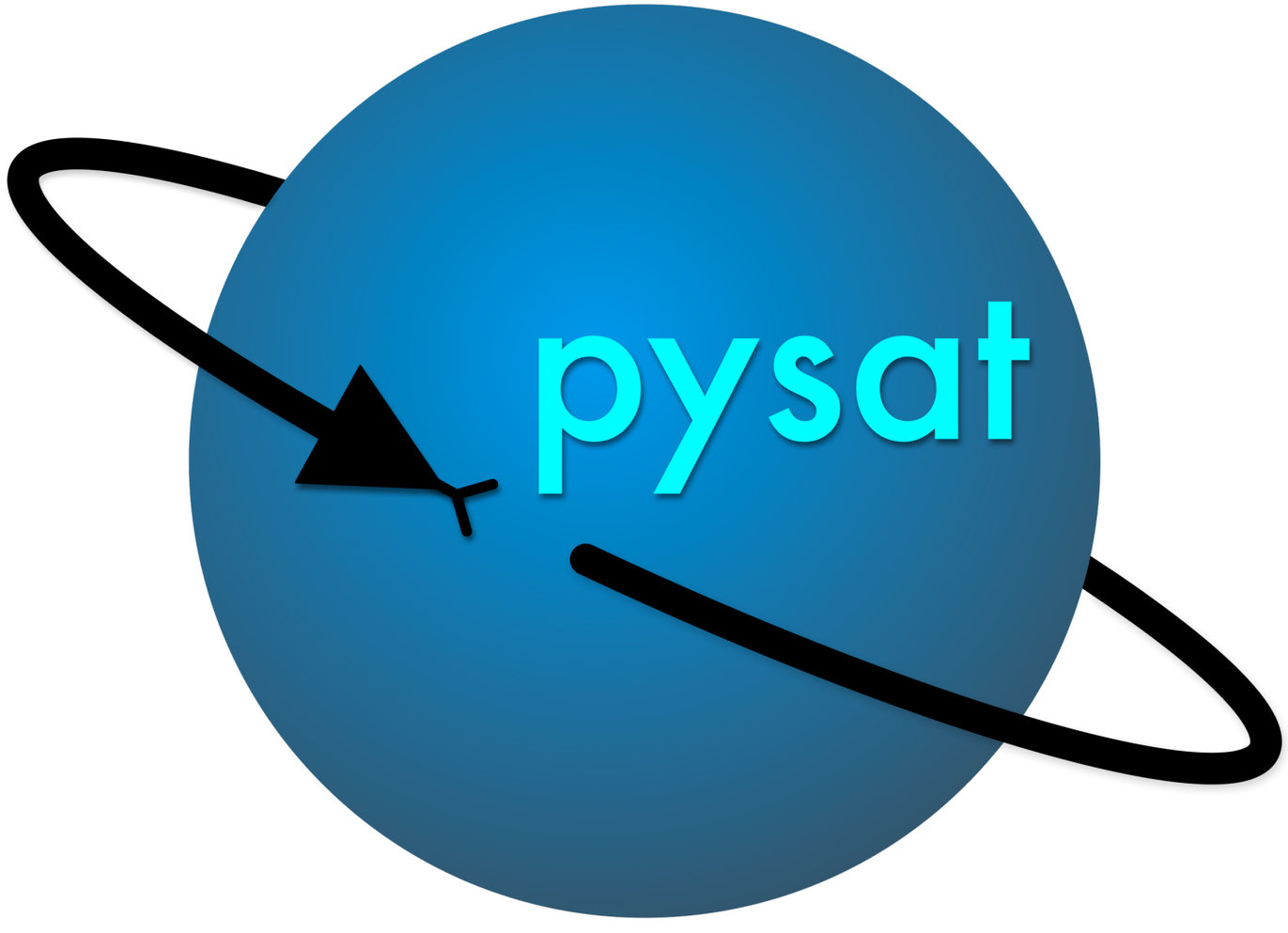
Python Satellite Data Analysis Toolkit

Russell Stoneback - University of Texas at Dallas
Jeff Klenzing - NASA
Angeline Burrell - University of Leicester



Overview

- High level package to load, download, and analyze space science data, regardless of instrument type
  - System for system science
- Instrument independent analysis functions
- Analysis is independent of file distribution
  - Supports orbit iteration with on the fly determination of orbit breaks
  - Loads by day independent of file data distribution

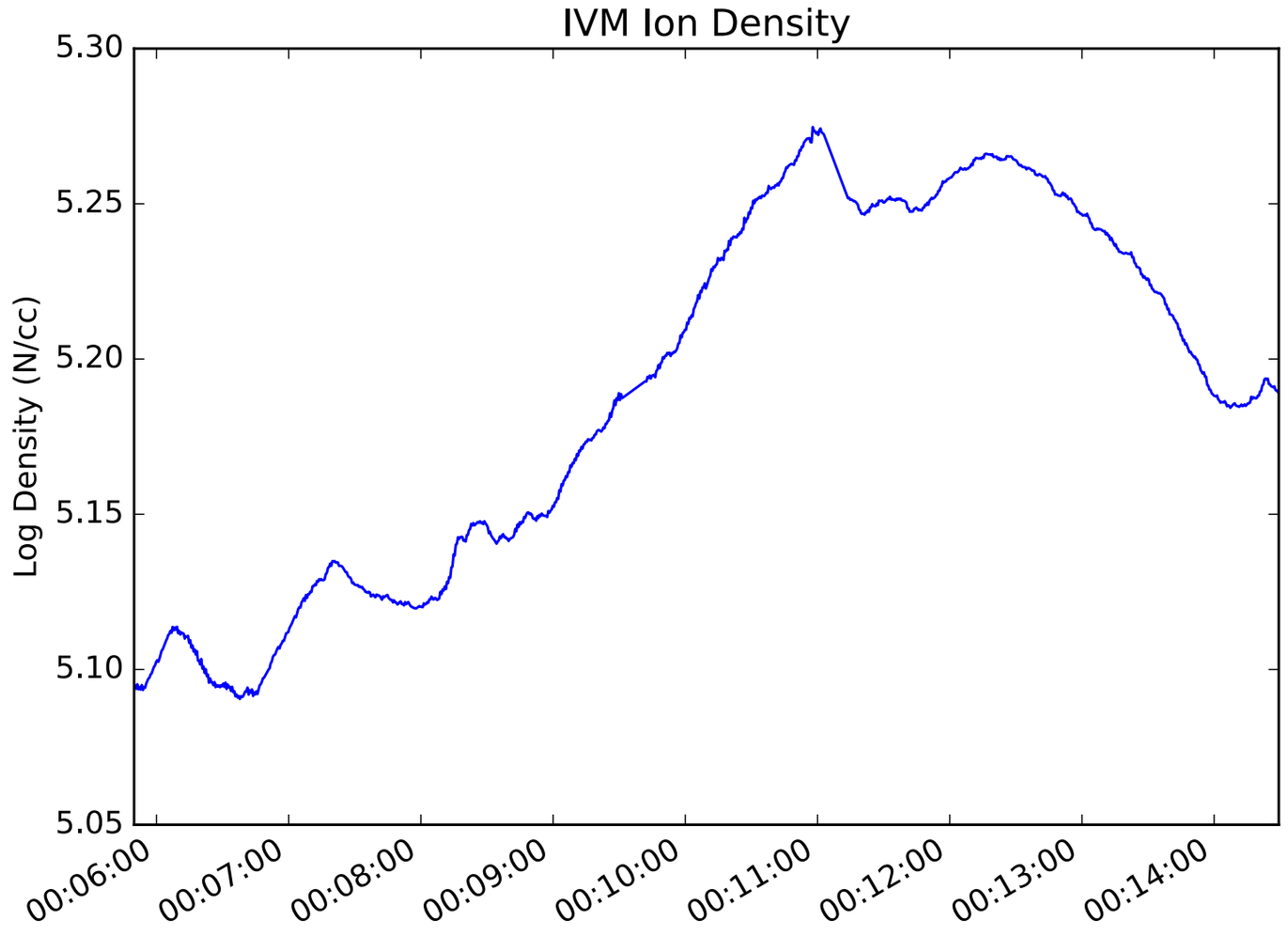
C/NOFS IVM

```
In [31]: import pysat
    ...: pysat.utils.set_data_dir('/Users/rstoneba/demo')
    ...: ivm = pysat.Instrument('cnofs', 'ivm', clean_level='clean')
    ...: ivm.download(pysat.datetime(2010,1,1), pysat.datetime(2010, 1, 2))
    ...: ivm.load(2010,1)
    ...: np.log10(ivm[0:1000,'ionDensity']).plot(title='IVM Ion Density')
    ...: plt.ylabel('Log Density (N/cc)')
Downloading data to:  /Users/rstoneba/demo/cnofs/ivm/
Downloading file for 01/01/10
Downloading file for 01/02/10
Updating pysat file list
pysat is searching for cnofs ivm files.
Found 2 of them.
Updating instrument object bounds.
Returning cnofs ivm data for 01/01/10
Out[31]: <matplotlib.text.Text at 0x123eb2790>
```

Data is preliminary

C/NOFS IVM

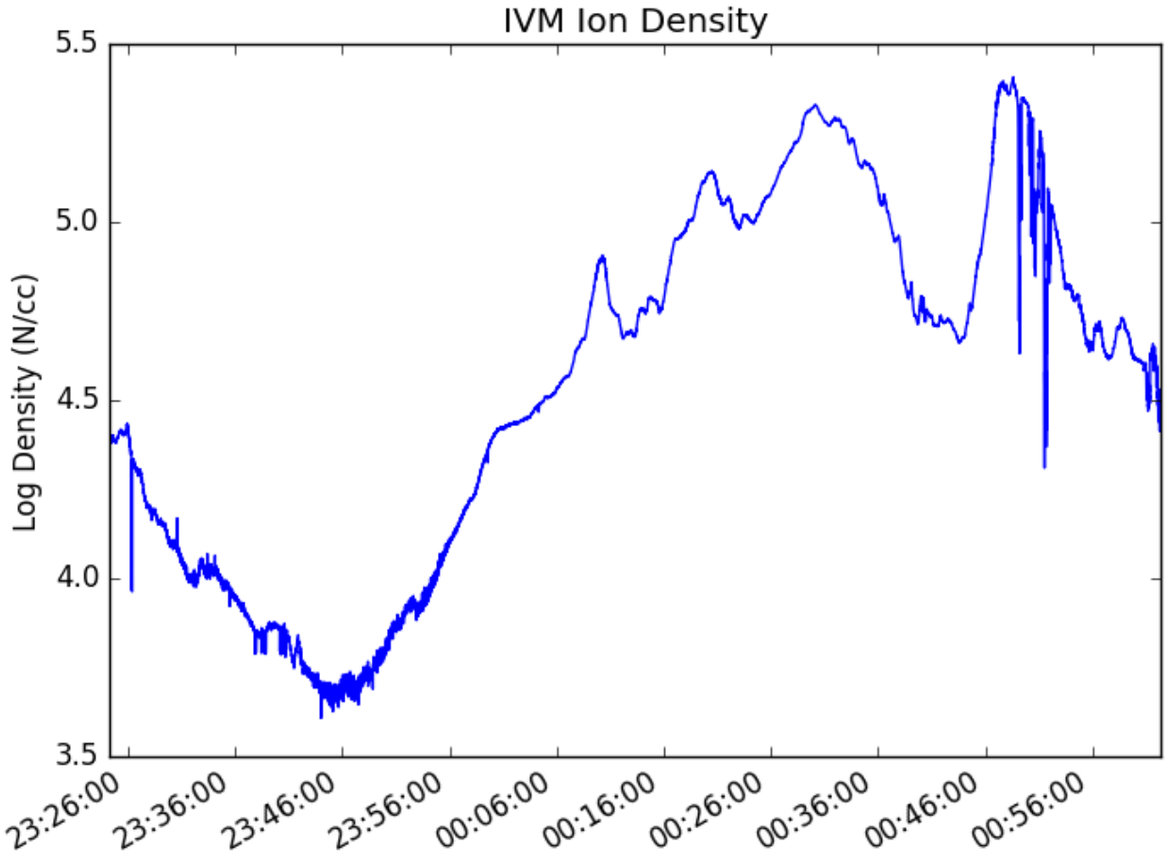
```
In [31]: import pysat
    ...: pysat.utils.set_data_dir('/Users/rstoneba/demo')
    ...: ivm = pysat.Instrument('cnofs', 'ivm', clean_level='clean')
    ...: ivm.download(pysat.datetime(2010,1,1), pysat.datetime(2010, 1, 2))
    ...: ivm.load(2010,1)
    ...: np.log10(ivm[0:1000,'ionDensity']).plot(title='IVM Ion Density')
    ...: plt.ylabel('Log Density
Downloading data to:  /Users/rsto
Downloading file for 01/01/10
Downloading file for 01/02/10
Updating pysat file list
pysat is searching for cnofs ivm
Found 2 of them.
Updating instrument object bounds
Returning cnofs ivm data for 01/0
Out[31]: <matplotlib.text.Text at
```



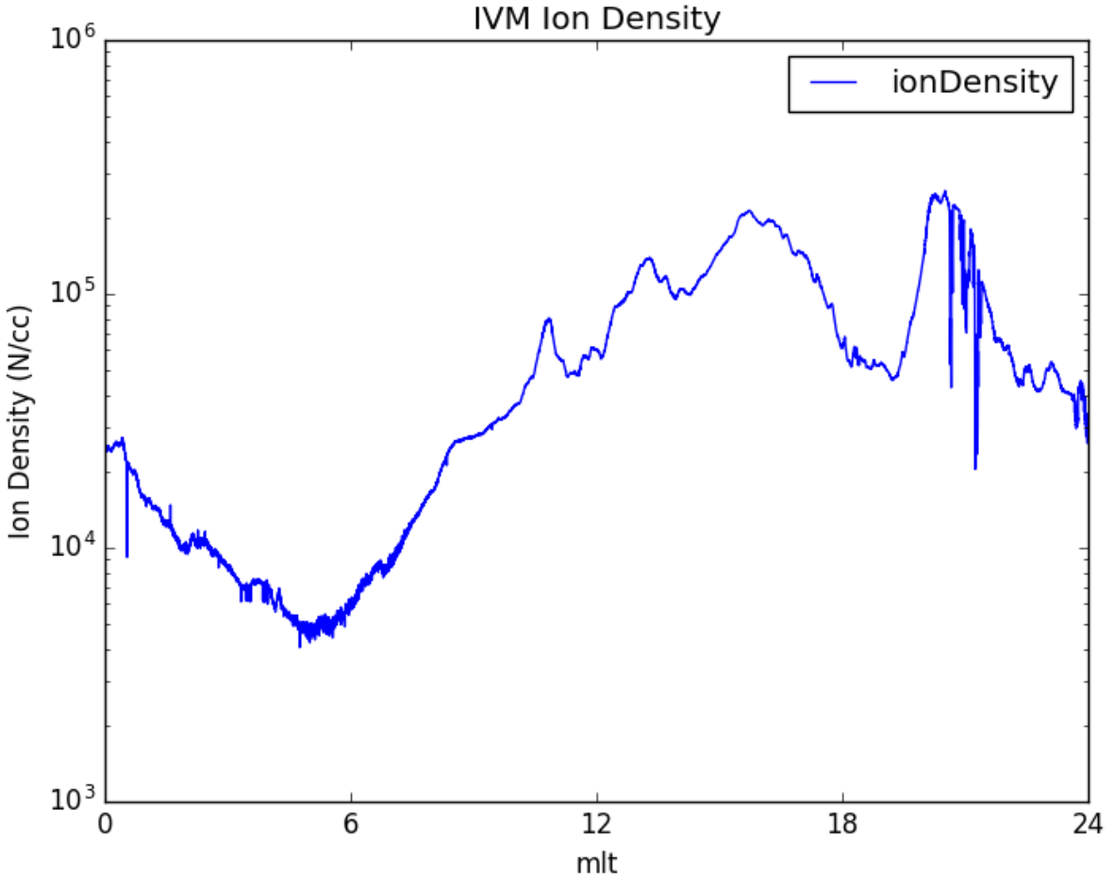
Data is preliminary

C/NOFS IVM by Orbit

```
In [40]: ivm = pysat.Instrument('cnofs', 'ivm',
    ...:                         clean_level='dirty',
    ...:                         orbit_info={'index':'mlt'})
    ...: ivm.load(2010,2)
    ...: ivm.orbits.next()
    ...: np.log10(ivm['ionDensity']).plot(title='IVM Ion Density')
    ...: plt.ylabel('Log Density (N/cc)')
Returning cnofs ivm data for 01/02/10
Returning cnofs ivm data for 01/01/10
Returning cnofs ivm data for 01/02/10
Loaded Orbit:0
Out[40]: <matplotlib.text.Text at 0x1250fab90>
```




```
In [48]: ivm = pysat.Instrument('cnofs', 'ivm',
    ...:                         clean_level='dirty',
    ...:                         orbit_info={'index':'mlt'})
    ...: ivm.load(2010,2)
    ...: ivm.orbits.next()
    ...: ivm.data.plot(x='mlt', y='ionDensity',
    ...:               title='IVM Ion Density',
    ...:               logy=True,
    ...:               xticks=[0,6,12,18,24])
    ...: plt.ylabel('Ion Density (N/cc)')
Returning cnofs ivm data for 01/02/10
Returning cnofs ivm data for 01/01/10
Returning cnofs ivm data for 01/02/10
Loaded Orbit:0
Out[48]: <matplotlib.text.Text at 0x1284af410>
```

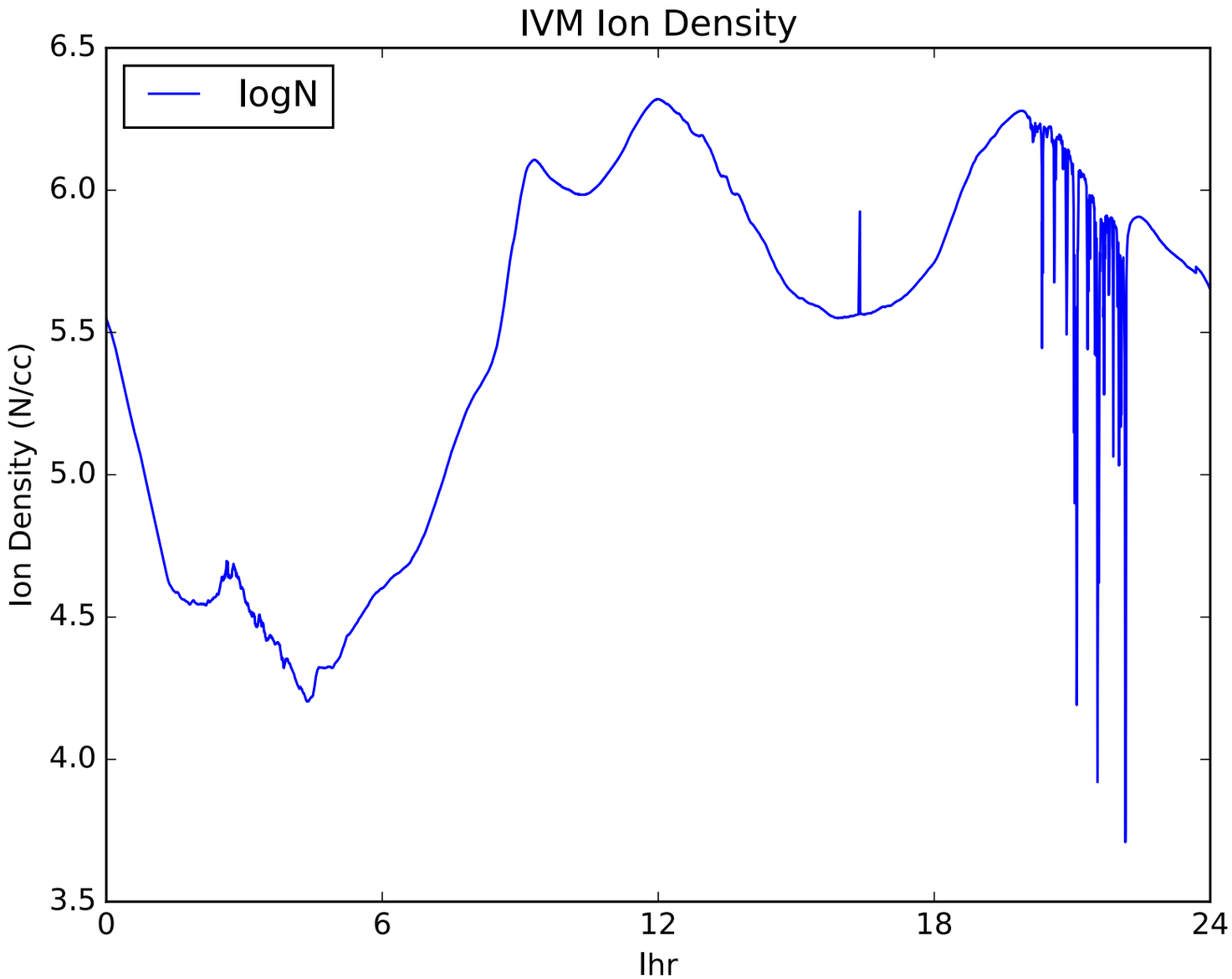


ROCSAT

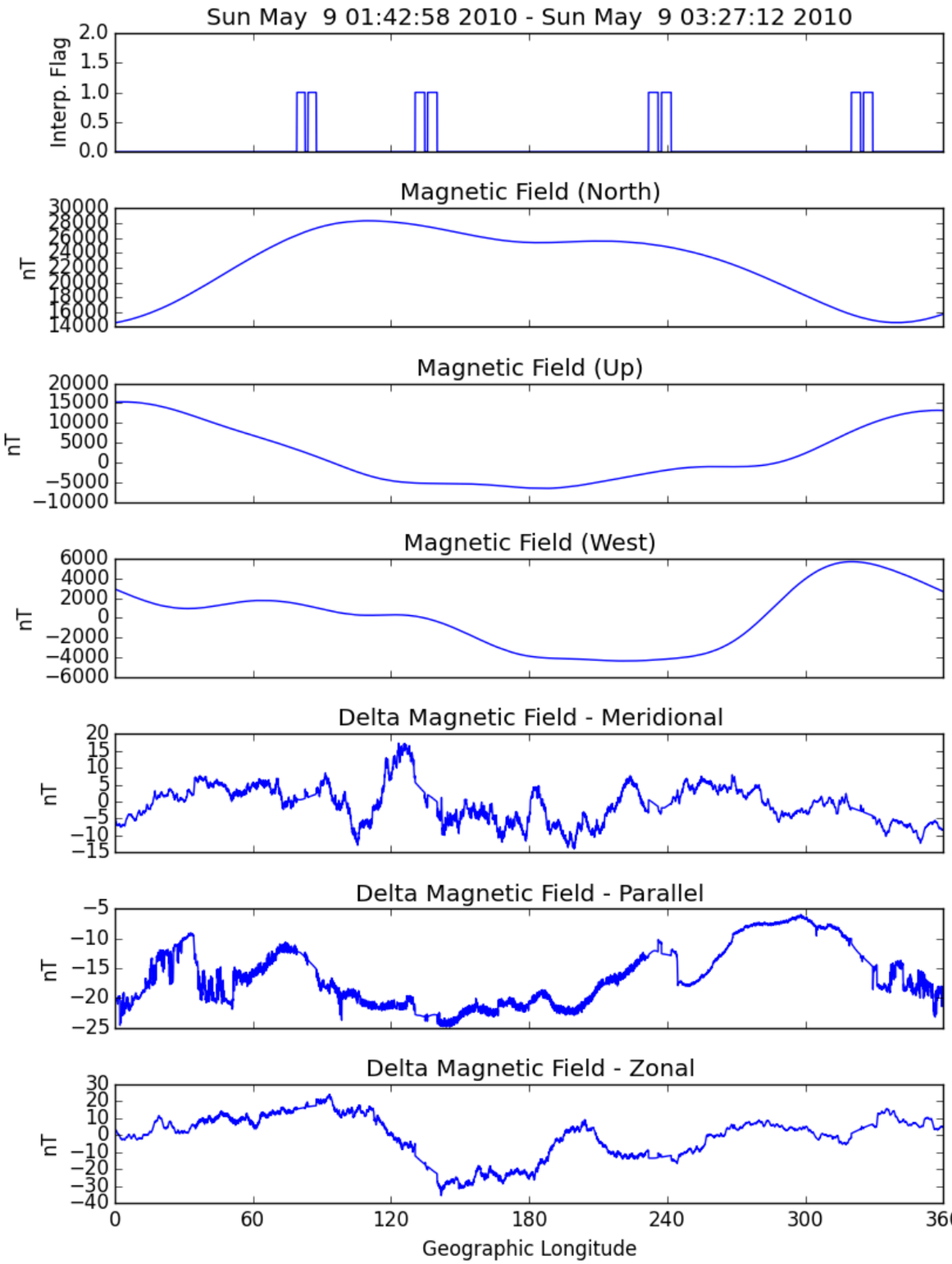
```
In [17]: ivm = pysat.Instrument('rocsat', 'ivm',
    ...:                         clean_level='none',
    ...:                         orbit_info={'index':'lhr'})
    ...: ivm.download(pysat.datetime(2002,1,1), pysat.datetime(2002,1,2))
    ...: ivm.load(2002,2)
    ...: ivm.orbits.next()
    ...: ivm.data.plot(x='lhr', y='logN',
    ...:               title='IVM Ion Density',
    ...:               xticks=[0,6,12,18,24])
    ...: plt.ylabel('Ion Density (N/cc)')
pysat is searching for rocsat ivm files.
Unable to find any files. If you have the necessary files please check pysat settings and file locations.
Downloading data to:  /Users/rstoneba/demo/rocsat/ivm/
Downloading file for 01/01/02
Downloading file for 01/02/02
Updating pysat file list
pysat is searching for rocsat ivm files.
Found 2 of them.
Updating instrument object bounds.
Returning rocsat ivm data for 01/02/02
Returning rocsat ivm data for 01/01/02
Returning rocsat ivm data for 01/02/02
Loaded Orbit:0
Out[17]: <matplotlib.text.Text at 0x124132e50>
```

ROCSAT

```
In [17]: ivm = pysat.Instrument('rocsat', 'ivm',
    ...:                         clean_level='none',
    ...:                         orbit_info={'index':'lhr'})
    ...: ivm.download(pysat.datetime(2002,1,1), pysat.datetime(2002,1,2))
    ...: ivm.load(2002,2)
    ...: ivm.orbits.next()
    ...: ivm.data.plot(x='lhr', y='log
    ...:               title='IVM Ion
    ...:               xticks=[0,6,12,
    ...: plt.ylabel('Ion Density (N/cc
pysat is searching for rocsat ivm file
Unable to find any files. If you have
Downloading data to:  /Users/rstoneba/
Downloading file for 01/01/02
Downloading file for 01/02/02
Updating pysat file list
pysat is searching for rocsat ivm file
Found 2 of them.
Updating instrument object bounds.
Returning rocsat ivm data for 01/02/02
Returning rocsat ivm data for 01/01/02
Returning rocsat ivm data for 01/02/02
Loaded Orbit:0
Out[17]: <matplotlib.text.Text at 0x1
```

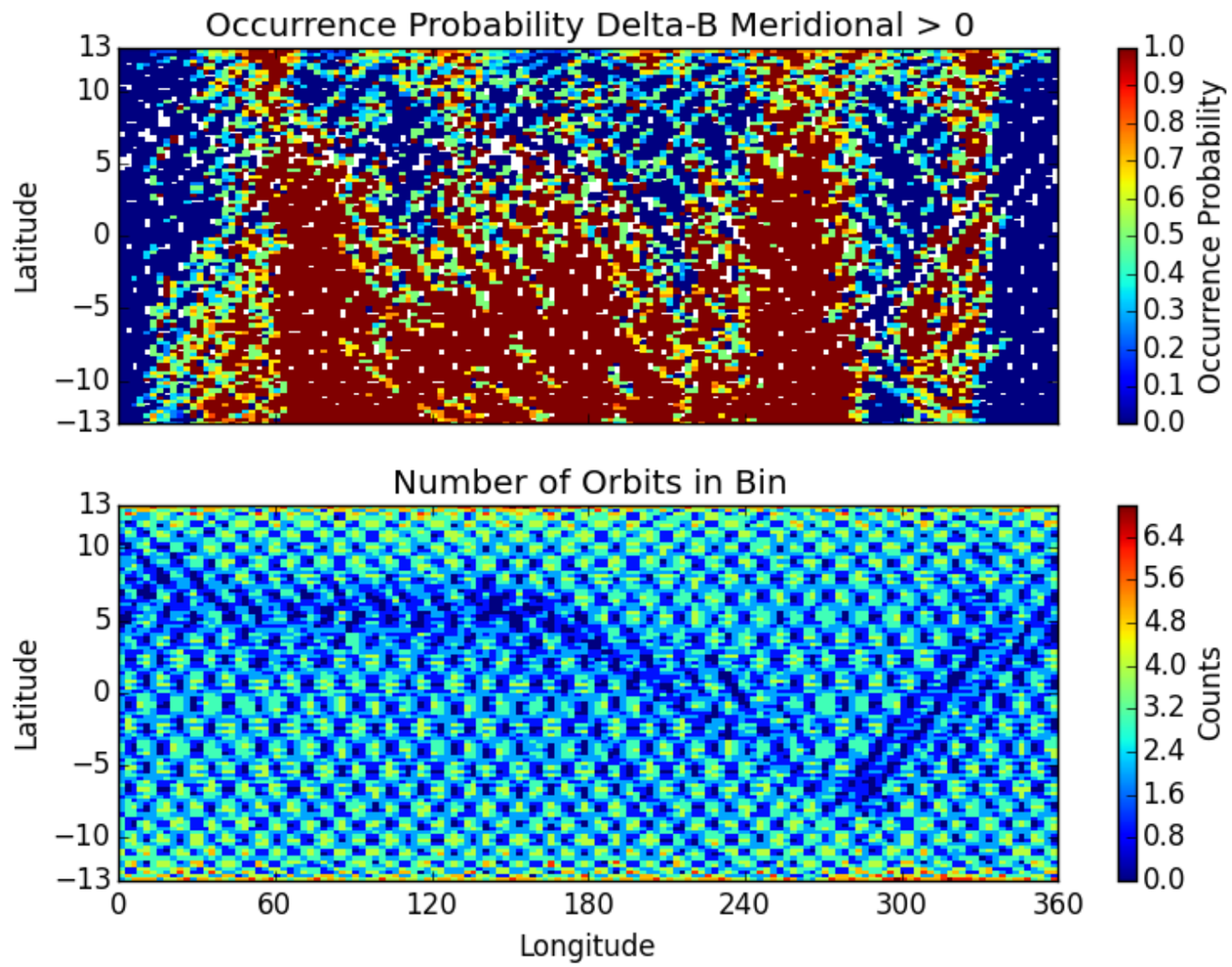


C/NOFS VEFI



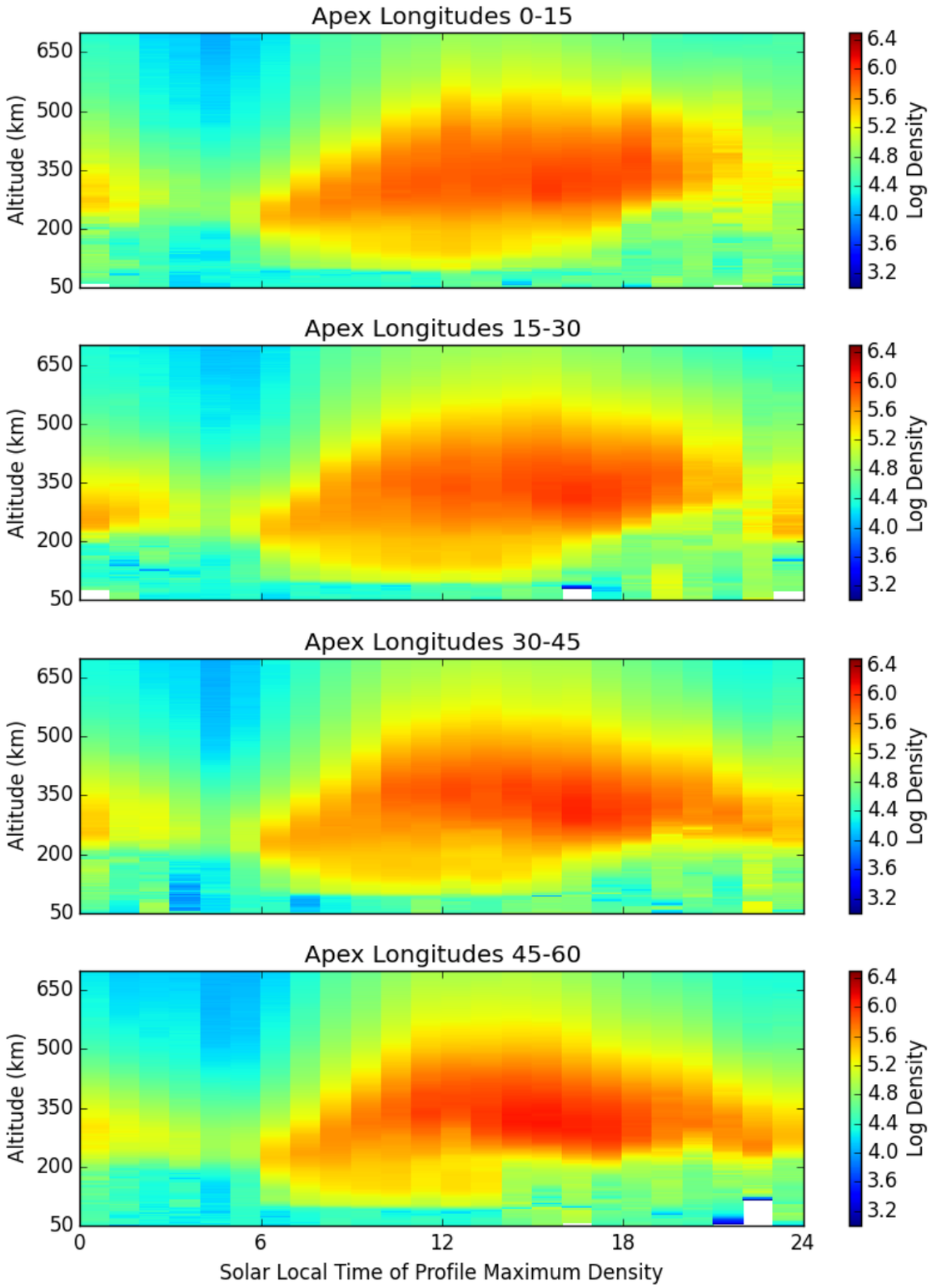

```
# select vefi dc magnetometer data, use longitude to determine where
# there are changes in the orbit (local time info not in file)
orbit_info = {'index':'longitude', 'kind':'longitude'}
vefi = pysat.Instrument(platform='cnofs', name='vefi', tag='dc_b',
                        clean_level=None, orbit_info=orbit_info)
```

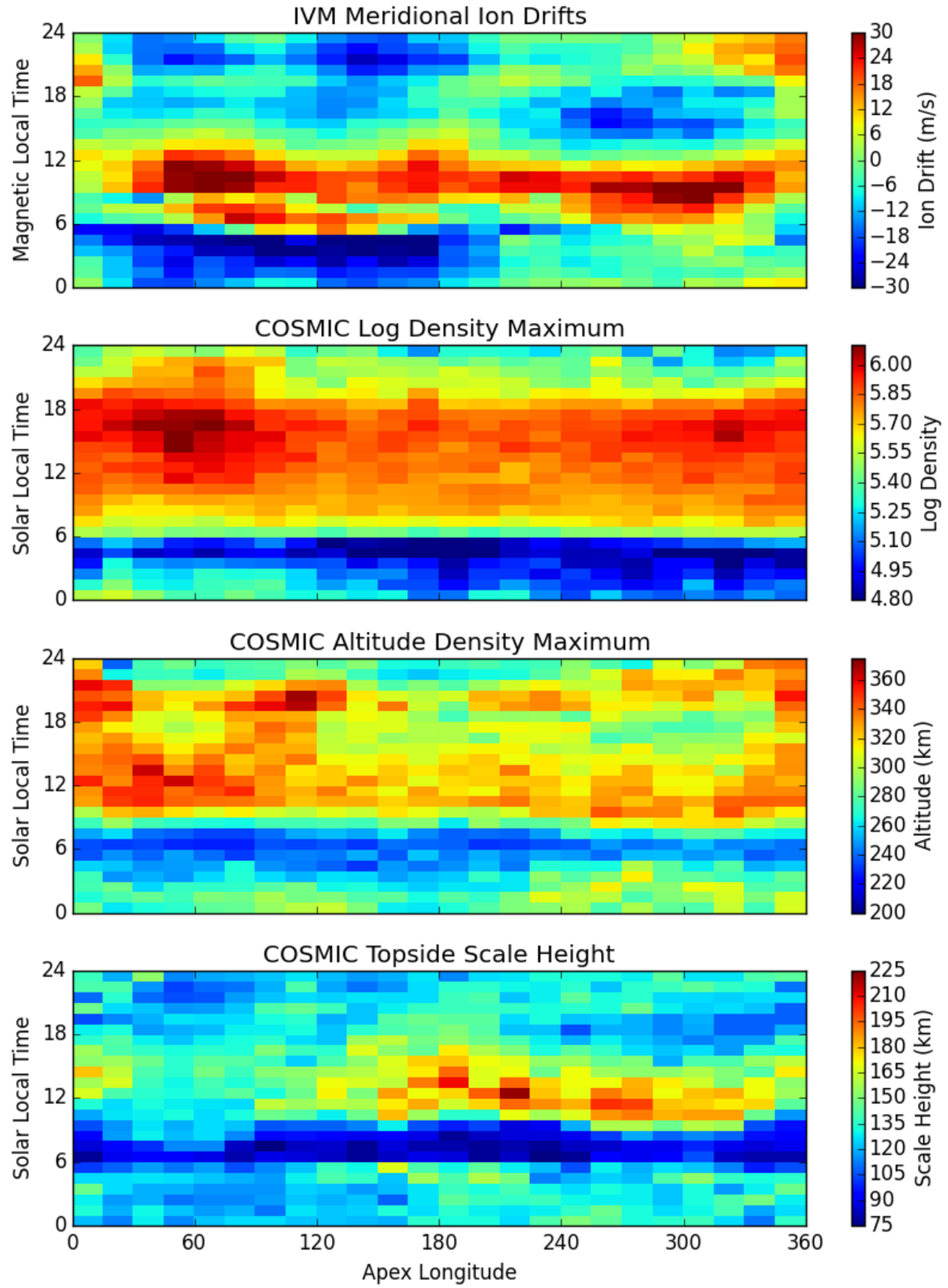
```
# perform occurrence probability calculation
# any data added by custom functions is available within routine below
ans = pysat.ssnl.occur_prob.by_orbit2D(vefi, [0,360,144], 'longitude',
                [-13,13,104], 'latitude', ['dB_mer'], [0.], returnBins=True)
```



Full Code in Demo Area of Repo

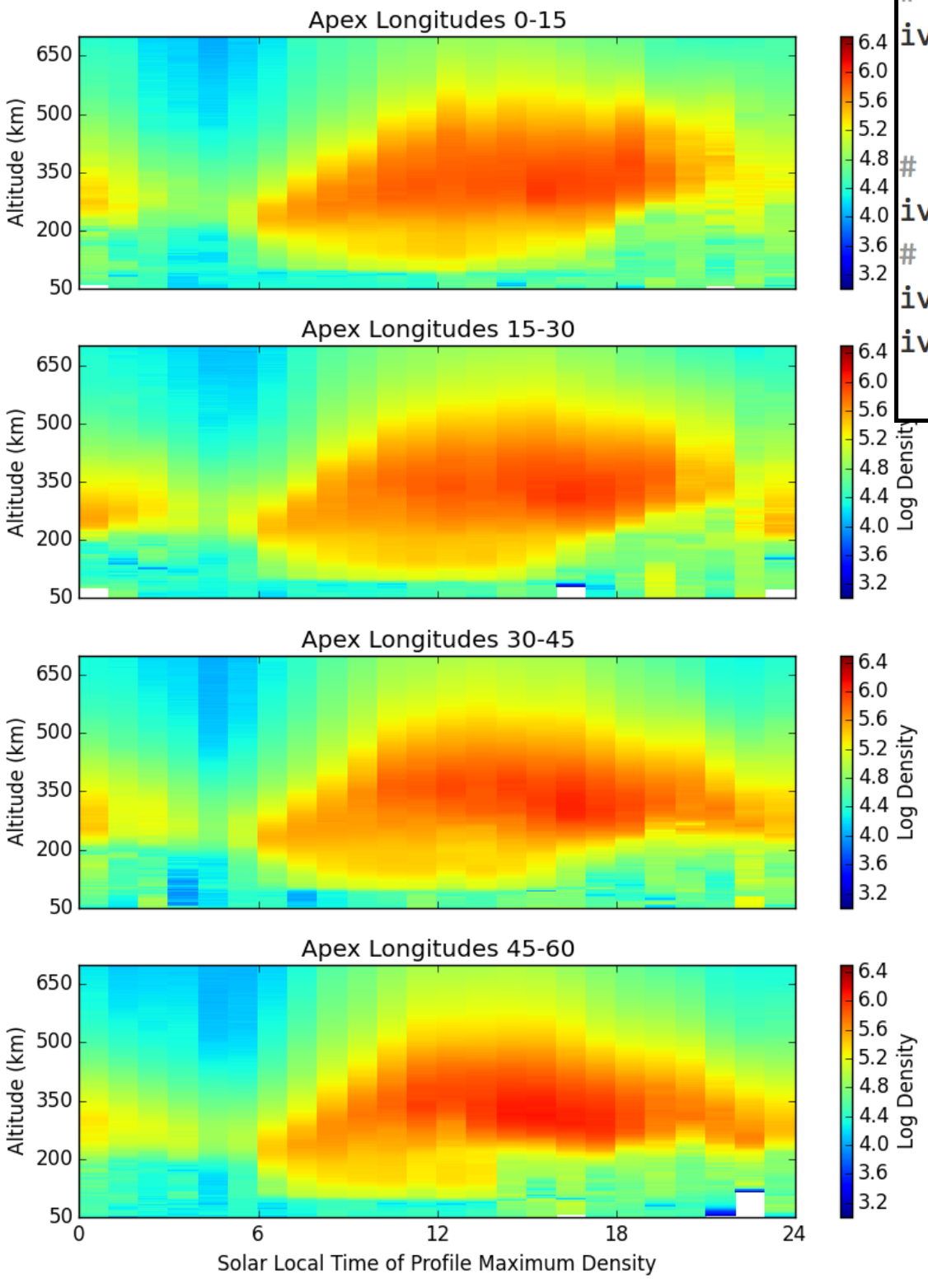
COSMIC and IVM Demo

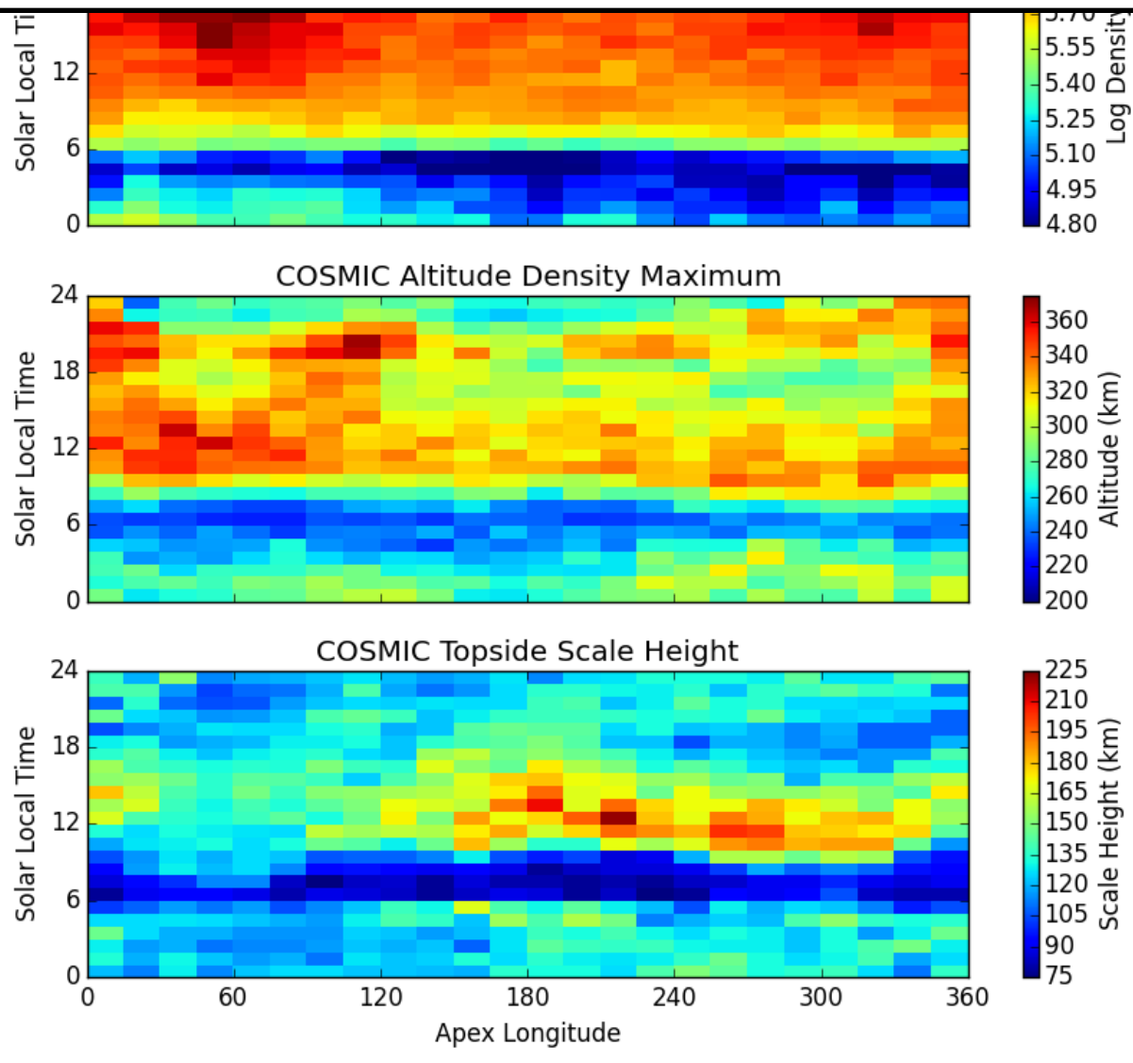




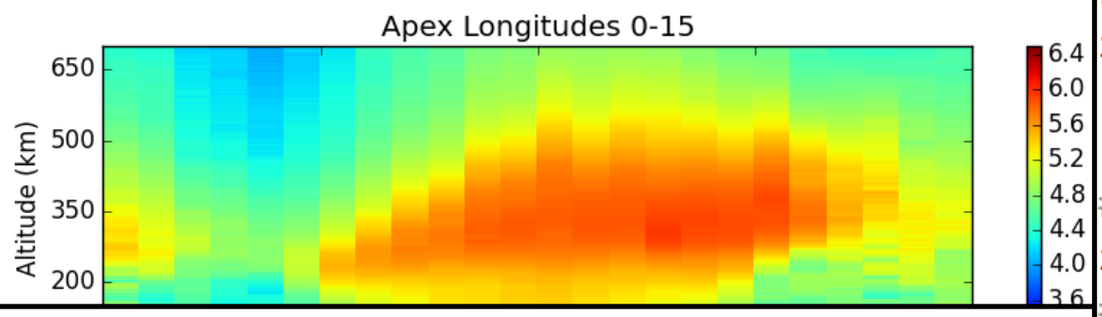
COSMIC and IVM Demo

```
# instantiate IVM Object
ivm = pysat.Instrument(platform='cnofs',
                       name='ivm', tag='',
                       clean_level='clean')
# restrict meausurements to those near geomagnetic equator
ivm.custom.add(restrictMLAT, 'modify', maxMLAT=25.)
# perform seasonal average
ivm.bounds = (startDate, stopDate)
ivmResults = pysat.ssnl.avg.median2D(ivm, [0,360,24], 'alon',
                       [0,24,24], 'mlt', ['ionVelmeridional'])
```



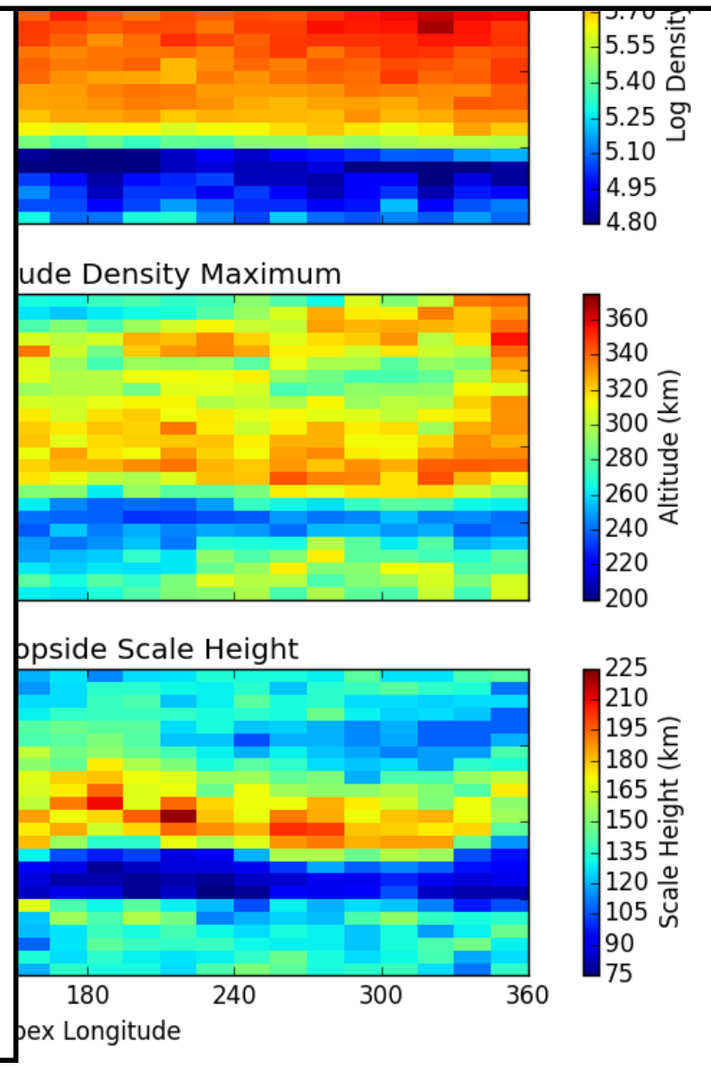


COSMIC and IVM Demo

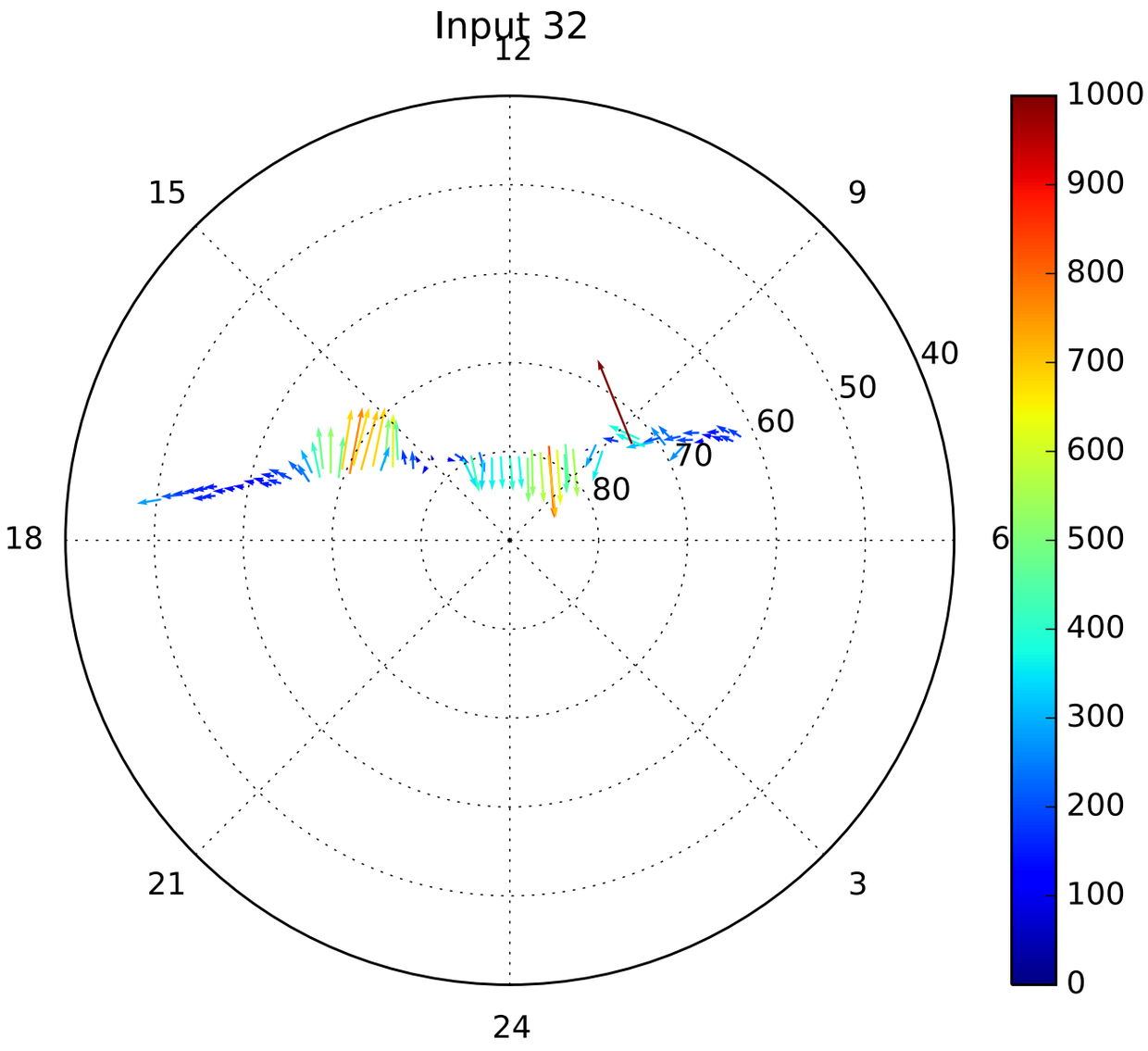


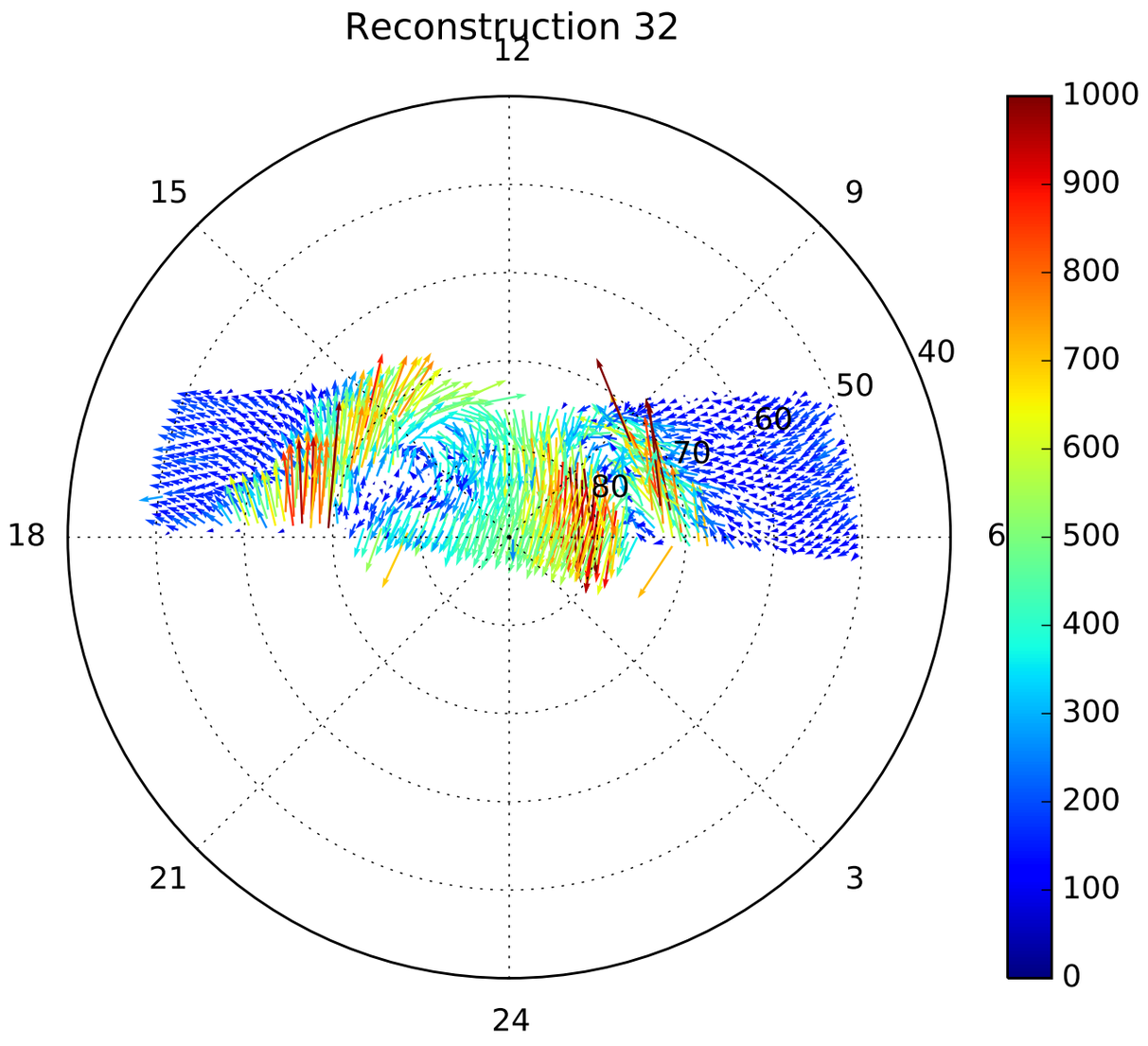

```
# create COSMIC instrument object
cosmic = pysat.Instrument(platform='cosmic2013',
                          name='gps',tag='ionprf'
                          clean_level='clean',
                          altitude_bin=3)
# apply custom functions to all data that is loaded through cosmic
cosmic.custom.add(addApexLong, 'add')
# select locations near the magnetic equator
cosmic.custom.add(filterMLAT, 'modify', mlatRange=(0.,10.) )
# take the log of NmF2 and add to the dataframe
cosmic.custom.add(addlogNm, 'add')
# calculates the height above hmF2 to reach Ne < NmF2/e
cosmic.custom.add(addTopsideScaleHeight, 'add')

# do an average of multiple COSMIC data products
# from startDate through stopDate
# a mixture of 1D and 2D data is averaged
cosmic.bounds = (startDate, stopDate)
cosmicResults = pysat.ssnl.avg.median2D(cosmic, [0,360,24], 'apex_long',
       [0,24,24],'edmaxlct', ['profiles', 'edmaxalt', 'lognm', 'thf2'])
```

```
# instantiate IVM Object
ivm = pysat.Instrument(platform='cnofs',
                       name='ivm', tag='',
                       clean_level='clean')
# restrict meausurements to those near geomagnetic equator
ivm.custom.add(restrictMLAT, 'modify', maxMLAT=25.)
# perform seasonal average
ivm.bounds = (startDate, stopDate)
ivmResults = pysat.ssnl.avg.median2D(ivm, [0,360,24], 'alon',
                    [0,24,24], 'mlt', ['ionVelmeridional'])
```



DMSP and DINEOFs

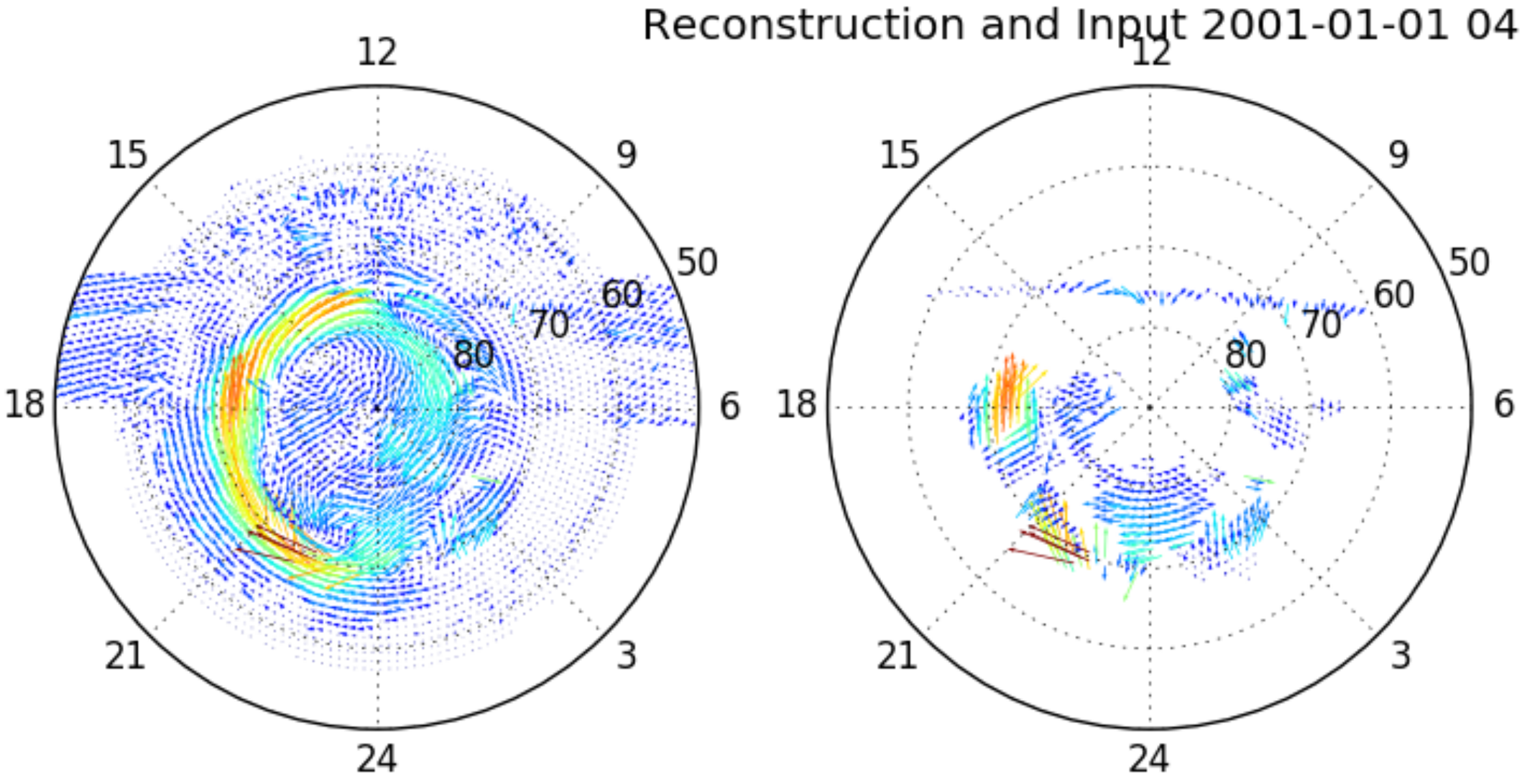




pysat used to integrate DMSP into DINEOFs
(Data-Based Assimilation)

National Science Foundation Grant 1259508

DMSP and SuperDARN



pysat used to integrate DMSP and SuperDARN into DINEOFs

National Science Foundation Grant 1259508

Current and Future Instruments

- C/NOFS IVM VEFI B-Field PLP
- COSMIC COSMIC-2013
- ROCSAT IVM SuperDARN (reqs. DaViTpy)
- KP Index OMNI-HRO
- CHAMP-STAR(partial) DMSP (private)
- ICON-IVM Ground Software ICON IVM
- COSMIC-2 IVM Ground Software COSMIC-2 IVM
- SORTIE IVM Ground Software SORTIE IVM

Adding NASA CDF

```
def list_files(tag=None, sat_id=None, data_path=None, format_str=None):
    """Return a Pandas Series of every file for chosen satellite data"""
    if format_str is None:
        format_str = 'rs_k0_ipei_{year:04d}{month:02d}{day:02d}_v01.cdf'
    if data_path is not None:
        return pysat.Files.from_os(data_path=data_path,
                                   format_str=format_str)
    else:
        raise ValueError ('A directory must be passed to the loading routine for IVM')


def load(fnames, tag=None, sat_id=None):
    import pysatCDF
    if len(fnames) <= 0 :
        return pysat.DataFrame(None), None
    else:
        with pysatCDF.CDF(fnames[0]) as cdf:
            return cdf.to_pysat()
```

pysat includes pysat instrument integration functions!

pysatCDF includes NASA library, available at terminal via:
pip install pysatCDF

Friendly CEDAR/GEM Challenge

Pls: Add your instrument to pysat

(via a grad student)

Max time cost is less than 1 graduate student week
(assumes programming experience but no python)

Suppose all instruments at CEDAR could be downloaded, loaded, modified, and analyzed with pysat.

- Common platform to distribute data and methods in support of publications at low cost (time) for both scientists and users
- Use of open and common tools sets standards for processing space science data
- Easier to analyze multiple instrument platforms
- Community would exceed current standards for public sharing

Thank you

- pysat
  https://github.com/rstoneback/pysat
  - Online Documentation with Tutorial
    http://rstoneback.github.io/pysat/
  - Demo Code and Reference Results
    https://github.com/rstoneback/pysat/tree/master/demo
  - Only some of the features shown today!
- pysatCDF
  https://github.com/rstoneback/pysatCDF
  - Includes NASA CDF library support
  - pip install pysatCDF