



DaViT-py

(Data Visualization Toolkit – Python)

M. Maimaiti ¹ and DaViT-py Team ²

¹ *Virginia Tech, Blacksburg, VA*

² Collaborators from different universities

What is DaViT-py?

- An open source space science analysis toolkit developed primarily in Python 2.7.
- Originally started at Virginia Tech.
- Now an active, international collaboration!

What is DaViT-py?

- Initially, it was designed to access and visualize SuperDARN data.
- Now, It is more than that...
- **Modules**
 - PyDARN: SuperDARN data access and plotting
 - GME: GeoMagnetic Environment, includes data from multiple sources (Kp, AE/AU/AL, A/SYMH, Dst, POES...)
 - Utils: various utilities for DaViT-py
 - Models: multiple space science models (IRI, IGRF, MSIS, Tsyganenko, AACGM) and ray-tracing

What is DaViT-py?

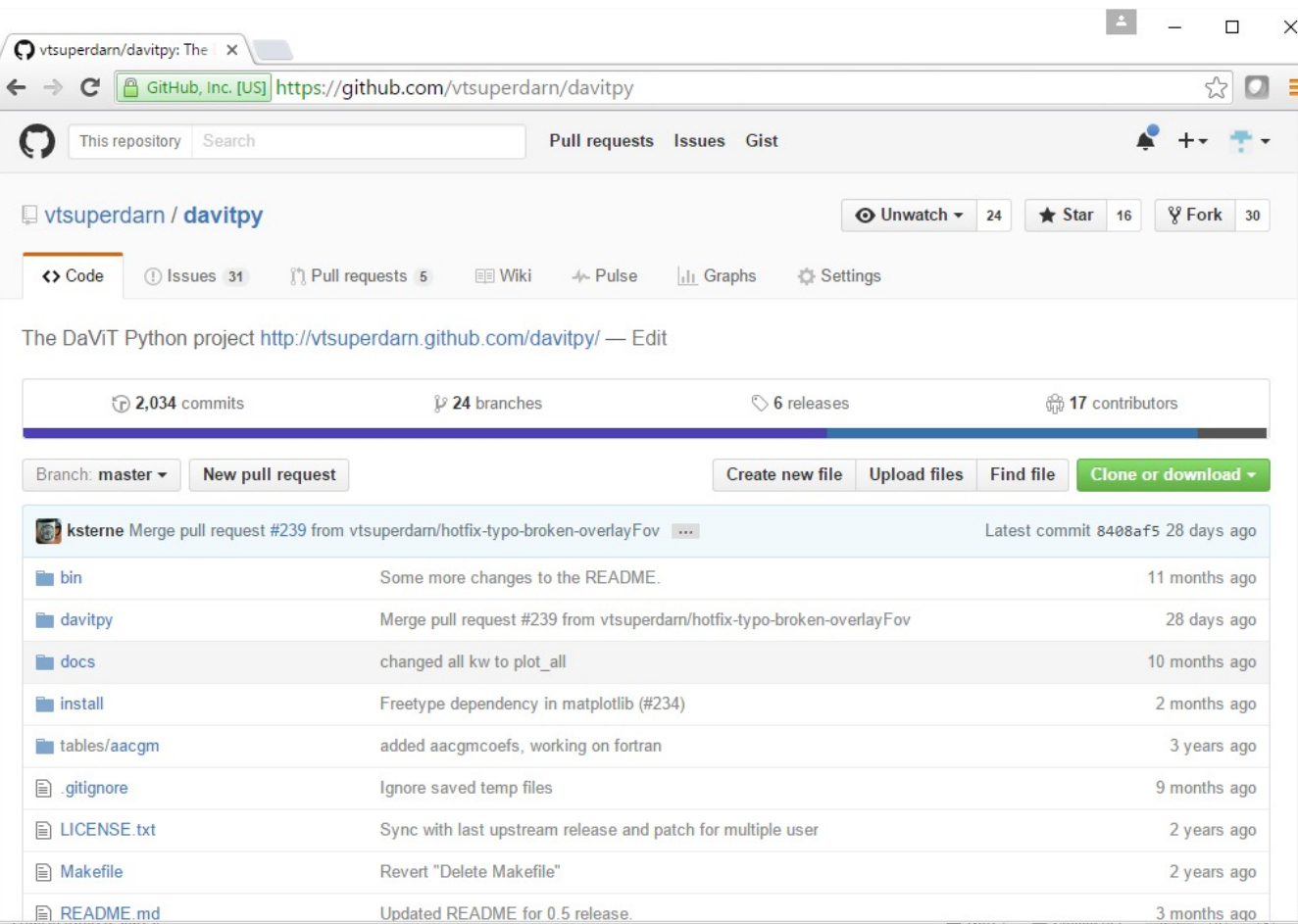
- DaViT-py has python wrappers for various fortran and c based models to assist in general space science studies.
 - **IRI** (International Reference Ionosphere 2011)
 - **IGRF** (International Geomagnetic Reference Field 2011)
 - **HF** Raytracing (SuperDARN ray tracing code coupled with IRI)
 - **MSIS** (Neutral atmosphere model)
 - **HWM** (Horizontal Wind Model 2014)
 - **Tsyganenko** (Geomagnetic Field Model)
 - **AACGM** (Altitude adjusted corrected geomagnetic model)

Current Collaborators

- Virginia Tech, USA
- University of Alaska Fairbanks, USA
- University of Saskatchewan, Canada
- University of Leicester, UK
- And others...

Collaboration Tools

Git and GitHub



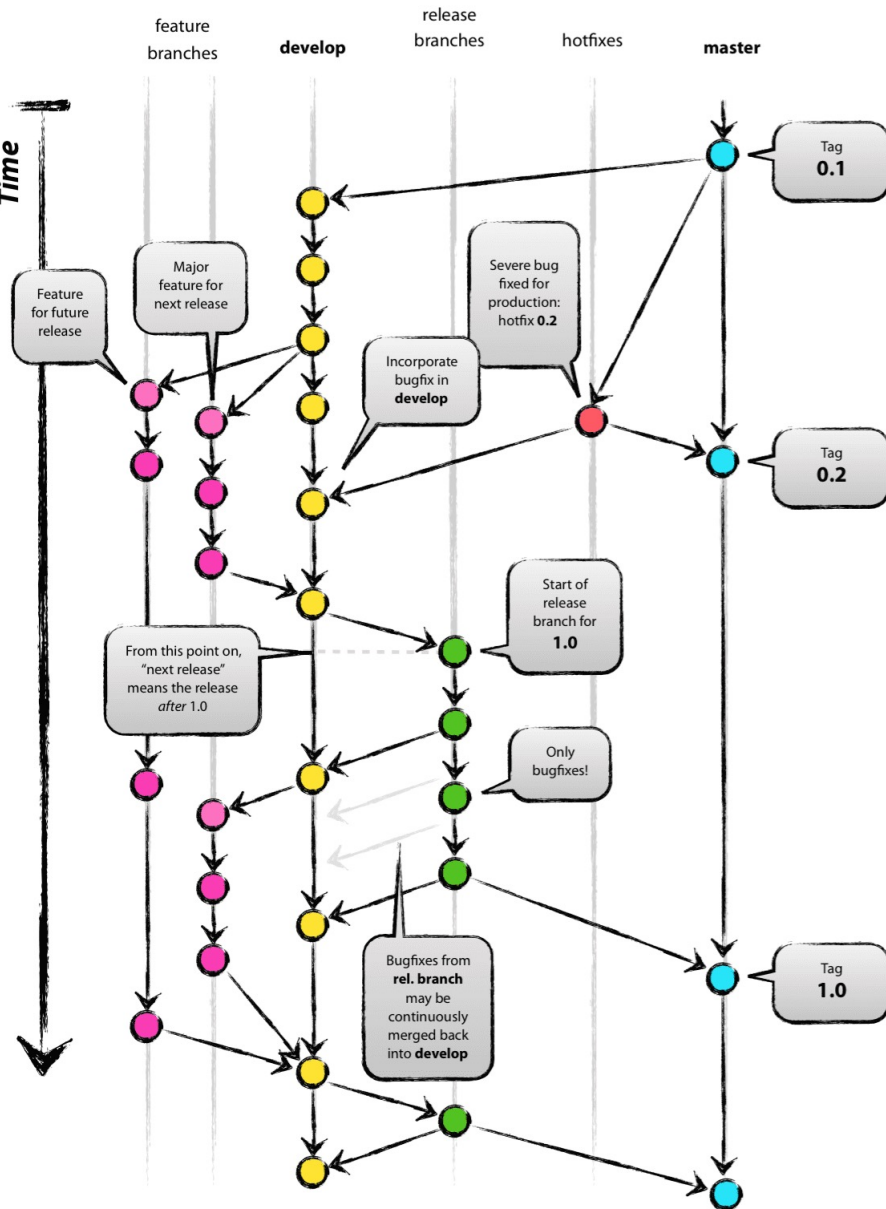
The screenshot shows the GitHub repository page for `vtsuperdam/davitpy`. The repository has 2,034 commits, 24 branches, 6 releases, and 17 contributors. The current branch is `master`. A list of files and folders is shown, including `bin`, `davitpy`, `docs`, `install`, `tables/aacgm`, `.gitignore`, `LICENSE.txt`, `Makefile`, and `README.md`. Each file has a commit message and a timestamp indicating when it was last updated.

- Version Control
- Bug Tracking
- Discussion
- Collaboration
- *You can go on this website right now and see everything...*

Development Model

- Set of rules everyone agrees with.
- Author: Vincent Driessen
- Original blog post: <http://nvie.com/posts/a-successful-git-branching-model>

- Feature branch
- Develop branch
- Release branch
- Master branch



DaViT-py Documentation

- Codes are Numpy PEP8 compliant
- Function arguments and returns are well documented
 - Doctrings right after each function definition
 - `<function name>?` *Or* `help(<function name>)`
- Example codes are provided for each function
 - Example codes as part of function doctings
 - Test codes in the end of module codes
 - Some examples in the form of Jupyter Notebooks that serve as tutorials

DaViT-py Documentation

MobaXterm

Terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Settings Help

Quick connect... 1.muhammad@sd-work4. x

```

49
50 def plotFan(sTime, rad, interval=60, fileType='fitex', param='velocity',
51             filtered=False, scale=[], channel=None, coords='geo',
52             colors='lasse', gsct=False, fov=True, edgeColors='face',
53             lowGray=False, fill=True, velscl=1000., legend=True,
54             overlayPoes=False, poesparam='ted', poesMin=-3., poesMax=0.5,
55             poesLabel=r"Total Log Energy Flux [ergs cm$^{2}$ s$^{-1}$]",
56             overlayBnd=False, show=True, png=False, pdf=False, dpi=500,
57             tFreqBands=[]):
58     """A function to make a fan plot
59
60     Parameters
61     -----
62     sTime : datetime
63         The start time you want to plot
64     rad
65         A list of 3 letter radar codes, e.g. ['bks'], e.g. ['bks','wal','gbr']
66     interval : Optional[int]
67         The the time period to be plotted, in seconds. default = 60
68     fileType : Optional[str]
69         The file type to plot, valid inputs are 'fitex','fitacf', 'lmfit'.
70         default = 'fitex'
71     param : Optional[str]
72         The parameter to be plotted, valid inputs are 'velocity', 'power',
73         'width', 'elevation', 'phi0'. default = 'velocity'
74     filtered : Optional[boolean]
75         A flag indicating whether the data should be boxcar filtered.
76         default = False
77     scale : Optional[list]
78         The min and max values of the color scale, i.e. [min,max]. If this is
79         set to [], then default values will be used
80     channel : Optional[char]
81         The channel for which to plot data. default = 'a'
82     coords : Optional[str]
83         The coordinate system to use; valid inputs are anything handled by
  
```

```

1 #-*- coding: utf-8 -*-
2 # Copyright (C) 2012 VT SuperDARN Lab
3 # Full license can be found in LICENSE.txt
4 #
5 # This program is free software: you can redistribute it and/or modify
6 # it under the terms of the GNU General Public License as published by
7 # the Free Software Foundation, either version 3 of the License, or
8 # (at your option) any later version.
9 #
10 # This program is distributed in the hope that it will be useful,
11 # but WITHOUT ANY WARRANTY; without even the implied warranty of
12 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
13 # GNU General Public License for more details.
14 #
15 # You should have received a copy of the GNU General Public License
16 # along with this program. If not, see <http://www.gnu.org/licenses/>.
17
18 """acf plotting module
19
20 A module for generating plotting ACF and XCF data
21
22 Module author ASR, 20141230
23
24 Functions
25 -----
26 plot_acf      Plot acf/xcf data
27 calc_blanked  Calc RX lag blanking
28 plot_rli     Plot range-lag-intensity
29 nuft         special Fourier transform
30 -----
31
32 """
33 import logging
34
35
  
```

fan.py 83,1 7% acfPlot.py 1,1 Top

"acfPlot.py" 644L, 21864C

[0] 0:vi* "sd-work4" 00:36 13-Jun-16

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <http://mobaxterm.mobatek.net>

Home x SuperDARN Data Pl... x

localhost:8888/notebooks/SuperDARN Data Plotting.ipynb

Most Visited Getting Started 82 useful Keyboard S... iPython Notebook Ke...

jupyter SuperDARN Data Plotting Last Checkpoint: 10/14/2014 (autosaved) Python 2

File Edit View Insert Cell Kernel Help

Cell Toolbar: None

This notebook will demonstrate how to do basic SuperDARN data plotting.

In [1]:

```
%pylab inline
import datetime
import os
import matplotlib.pyplot as plt
from davitpy import pydarn
```

Populating the interactive namespace from numpy and matplotlib

In [2]:

```
sTime = datetime.datetime(2008,2,22)
eTime = datetime.datetime(2008,2,23)
radar = 'bks'
beam = 7
```

Remote File RTI Plots

In [3]:

```
#The following command will print the docstring for the plotRti routine:
#pydarn.plotting.rti.plotRti?
```

In [4]:

```
fig = plt.figure(figsize=(14,12)) #Define a figure with a custom size.
pydarn.plotting.rti.plotRti(sTime, radar, eTime=eTime, bmnum=beam, figure=fig)
plt.show()
```

Found cached file: /tmp/sd/20080222.000000.20080223.000000.bks.fitex

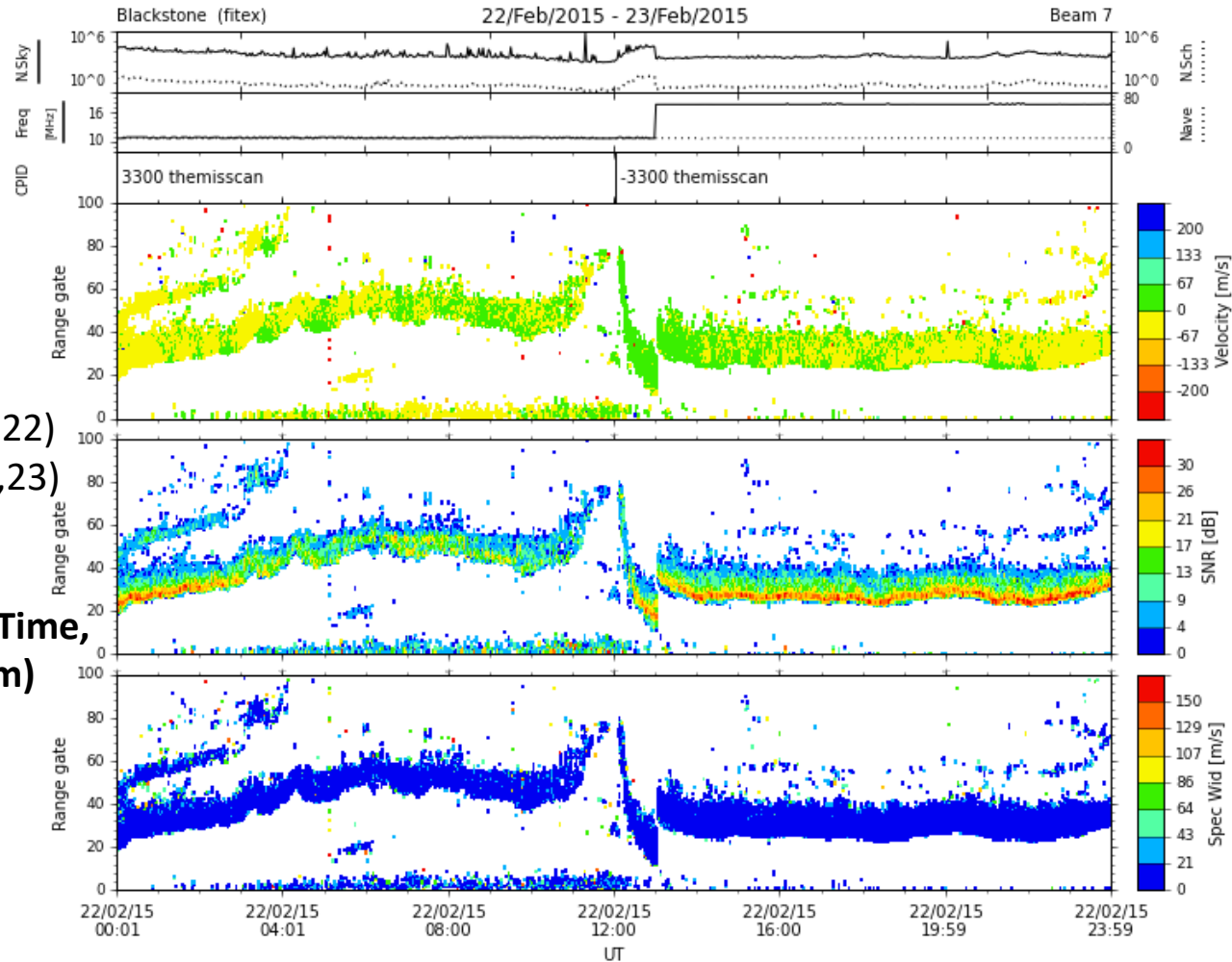
reached end of data
plotting took: 0:00:05.204070

/usr/local/lib/python2.7/dist-packages/matplotlib/figure.py:387: UserWarning: matplotlib is currently using a non-GUI backend, so cannot show the figure
"matplotlib is currently using a non-GUI backend, "

DaViT-py Demos: RTI Plot

```

import datetime
import os
import matplotlib.pyplot as plt
from davitpy import pydarn
sTime = datetime.datetime(2008,2,22)
eTime = datetime.datetime(2008,2,23)
radar = 'bks'
beam = 7
figs = pydarn.plotting.rti.plot_rti(sTime,
radar, eTime=eTime, bmnum=beam)
fig = figs[0]
fig.show()
  
```

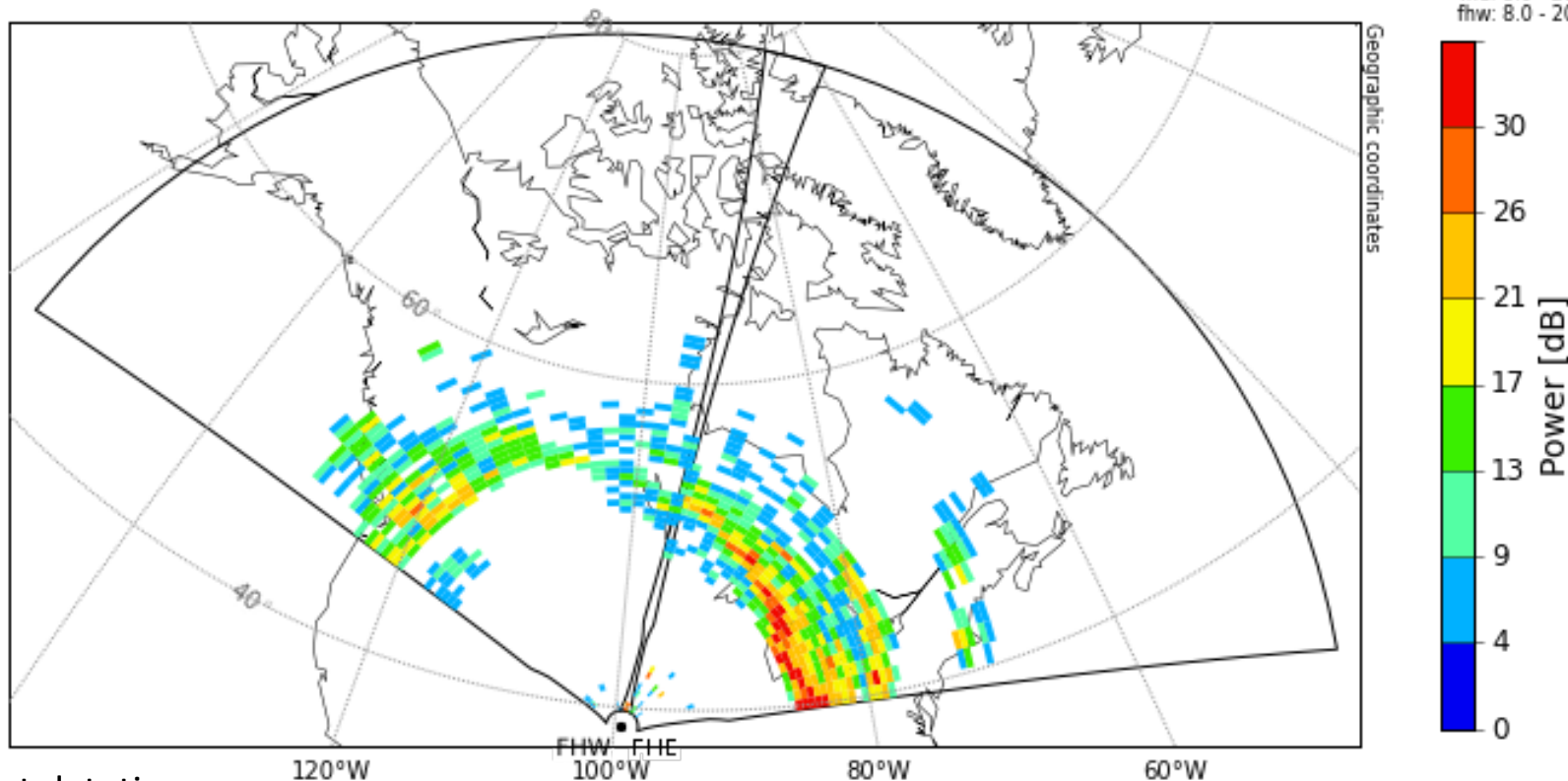


DaViT-py Demos: Fan Plot

[fitex]

2013/03/16

16:30 - 16:31

 Frequency filters:
 fhe: 8.0 - 20.0 MHz
 fhw: 8.0 - 20.0 MHz


```

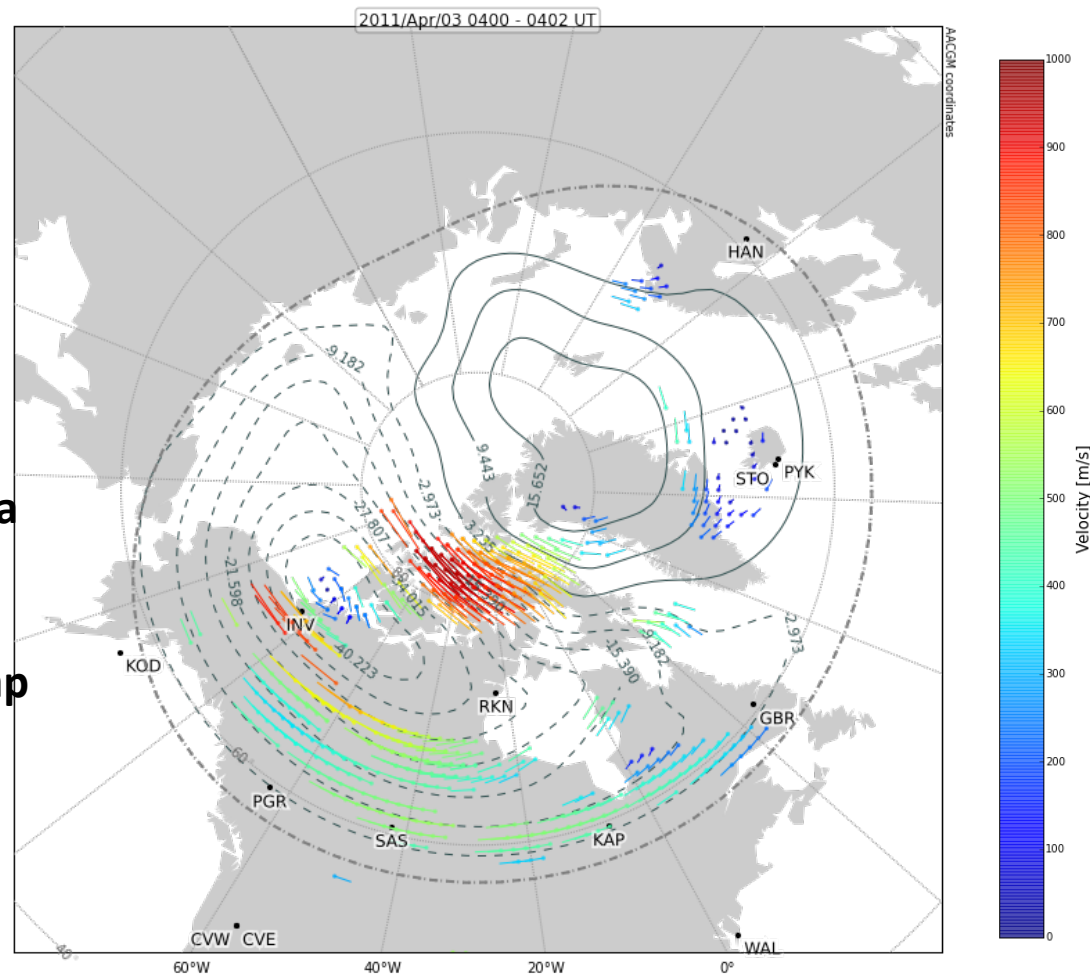
import datetime
import os
import matplotlib.pyplot as plt
from davitpy import pydarn
pydarn.plotting.fan.plotFan(datetime.datetime(2013,3,16,16,30),['fhe','fhw'],param='power',gsct=False)
  
```


DaViT-py Demos: Convection Map

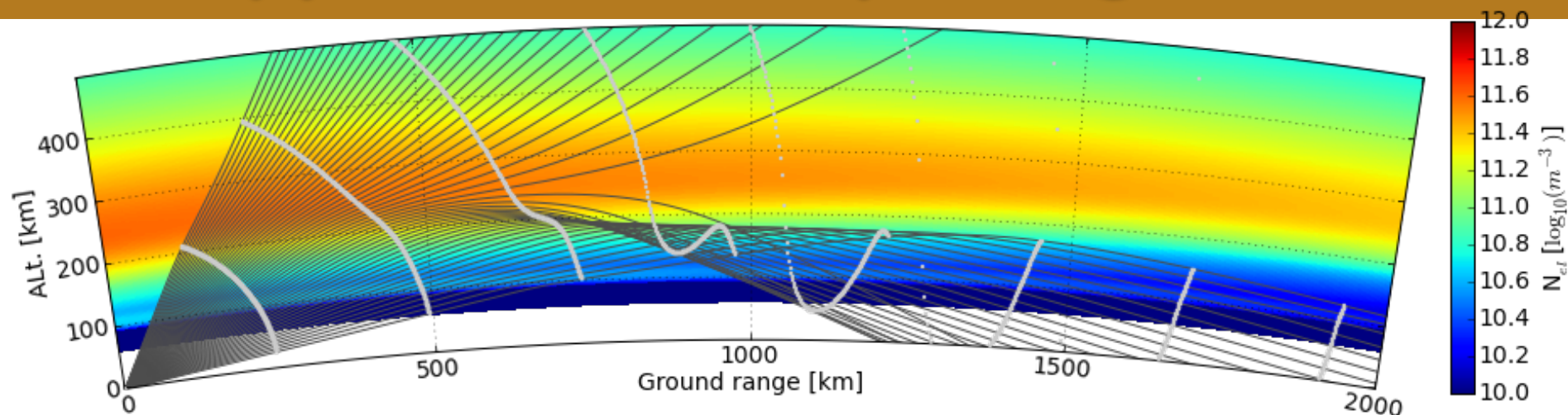
```

import datetime
import matplotlib.pyplot as plt
import
davitpy.pydarn.plotting.plotMapGrd
from davitpy.utils import *
fig = plt.figure(figsize=(15,15))
ax = fig.add_subplot(111)
sdate = datetime.datetime(2011,4,3,4,0)
mObj =
plotUtils.mapObj(boundinglat=50.,gridLa
bels=True, coords='mag')
mapDatObj =
davitpy.pydarn.plotting.plotMapGrd.Map
Conv(sdate, mObj, ax)
mapDatObj.overlayMapFitVel()
mapDatObj.overlayCnvCntrs()
mapDatObj.overlayHMB()

```



DaViT-py Demos: HF Raytracing



```

import datetime as dt
from davitpy.models import raydarn
sTime = dt.datetime(2006, 2, 22, 22); eTime = dt.datetime(2006, 2, 23, 5)
radar = 'wal' ; beam = 3; freq = 10.5
rto = raydarn.RtRun(sTime, eTime, rCode=radar, beam=beam, freq=freq, outDir='/tmp', nprocs=4)
rto.readRays()
figure(figsize=(15,5))
ax, aax, cbax = rto.ionos.plot(sTime)
ax, aax, cbax = rto.rays.plot(sTime, step=10, ax=ax, aax=aax)
rto.rays.showRange()
ax.grid()

```

DaViT-py Installation

- Detailed installation instruction can be found from:
 - <https://github.com/vtsuperdarn/DaViT-py>
 - README.md file in the DaViT-py package
- OS requirement
 - Most active development, testing, and use occurs in the Linux environment.
 - Macintosh install scripts are provided
 - We do not currently offer any Windows support, although **you are more than welcome to try and make it work in any environment you choose.**

Summary

- DaViT-py is mainly designed to conveniently access and visualize SuperDARN data, but it also includes other relevant space physics/space weather data sets and models.
- Open source collaborative projects have been proven to be very successful
 - Linux, Mozilla, Apache, Et al.
- Please join us!!

DaViT-py Related Links

- Have a look at the projects main page (public face): <http://vtsuperdarn.github.com/DaViT-py/>
- You can also find tutorials and other information on our github wiki: <http://github.com/vtsuperdarn/DaViT-py/wiki>
- Please join our development Google group, DaViT-py-dev (<https://groups.google.com/forum/#!forum/DaViT-py>).
- Please report any problems/comments using the Issues tab of the DaViT-py GitHub page, or use this link: <https://github.com/vtsuperdarn/DaViT-py/issues>
- Virginia Tech SuperDARN Page <http://vt.superdarn.org>