

py DARN

| By Marina Schmidt



UNIVERSITY OF
SASKATCHEWAN



- What is pyDARN?
- User Tutorials
 - Installation
 - IO/Summary plots
 - Documentation
 - How to Cite
 - Future work
 - Issues
- Developers - How to help
 - Testing Code
 - Code Reviews
 - Style Guide
 - Pull Requests



What is pyDARN?

pyDARN is a python library for visualizing SuperDARN data.

Mission Statement: **pyDARN** is a python library providing a *user friendly experience* while meeting *scientific needs*, all by following *best programming practices*:

- Flexibility
- Extendibility
- Low maintenance

The **Super Dual Auroral Radar Network (SuperDARN)** is a global network of (~36) scientific radars monitoring conditions in the near-Earth space environment.



Tutorial: Installation

```
$ pip3 install pydarn
```

`pyDARN` is on PyPi meaning it is pip installable.

! Warning: `pyDARN` requires `python 3.6+`

I highly encourage installing in a virtual environment either by python or conda to avoid system configuration issues.

```
$ python3 -m pip install --user virtualenv  
$ python3 -m virtualenv <virtual environment name>  
$ source <virtual environment name>/bin/activate  
$ pip3 install pydarn
```



UNIVERSITY OF
SASKATCHEWAN





Tutorial: Getting Data

You can download SuperDARN data via:



globus



**British
Antarctic Survey**

NATURAL ENVIRONMENT RESEARCH COUNCIL

Please contact myself or Kevin Krieger for access via Globus. Work on DOI'ing SuperDARN RAWACF and FITACF level data on FRDR is underway.



UNIVERSITY OF
SASKATCHEWAN





Tutorial: IO

```
import pydarn
```

```
fitacf_file = '20180220.C0.rkn.stream.fitacf'
```

```
# read an unzipped file
```

```
reader = pydarn.SDarnRead(fitacf_file)
```

```
records = reader.read_fitacf()
```

```
import bz2
```

```
import pydarn
```

```
# read in compressed file
```

```
fitacf_file = '20180220.C0.rkn.stream.fitacf.bz2'
```

```
with bz2.open(fitacf_file) as fp:
```

```
    fitacf_stream = fp.read()
```

```
# pyDARN functions to read a fitacf file stream
```

```
reader = pydarn.SDarnRead(fitacf_stream, True)
```

```
records = reader.read_fitacf()
```





Tutorial: Borealis IO

If you get Borealis data from SuperDARN Canada and want to convert the hdf5 to DMAP:

```
import pydarn
my_borealis_array_file = "path/to/file"
sdarn_file = "path/to/write/to"

converter = pydarn.BorealisConvert(my_borealis_file, "rawacf",
                                   sdarn_file, 0, borealis_file_structure='array')
```



UNIVERSITY OF
SASKATCHEWAN





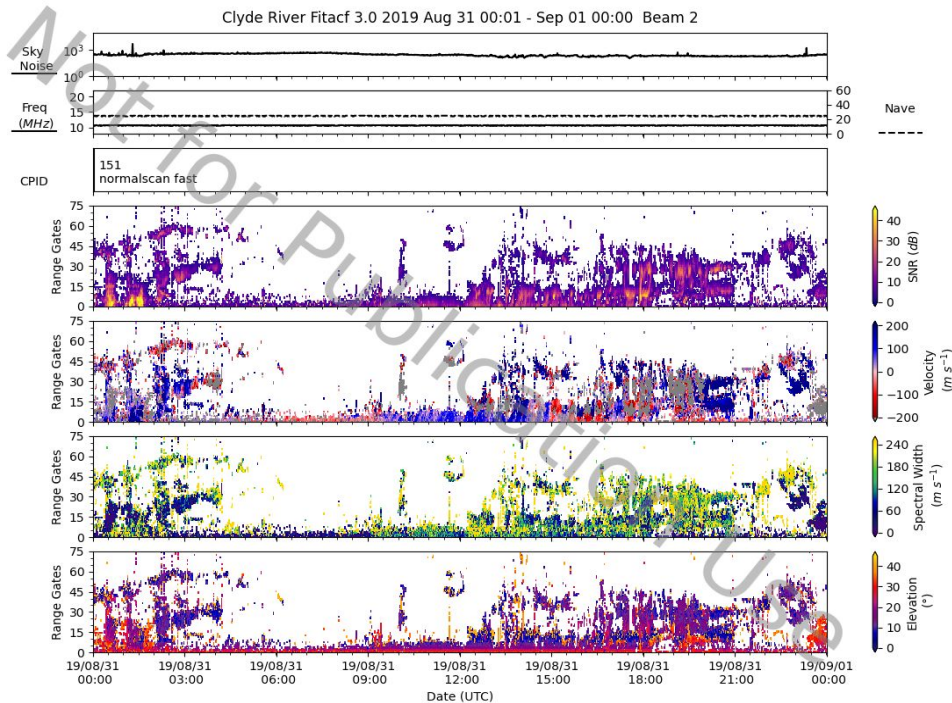
Tutorial: Summary Plots

```
import pydarn
import matplotlib.pyplot as plt

fitacf_file = "20190831.C0.cly.fitacf"
darn_read =
    pydarn.SDarnRead(fitacf_file)
fitacf_data =
    darn_read.read_fitacf()

pydarn.RTP.plot_summary(fitacf_data,
                        beam_num=2)

plt.show()
```



UNIVERSITY OF
SASKATCHEWAN





Tutorial: Hardware Files

```
# Read Saskatoon's hardware file  
hdw_data = pydarn.read_hdw_file('sas')  
print(hdw_data.geographic)
```

```
_Coord(lat=52.16, lon=-106.53, alt=494.0)
```

```
# Access Prince George's Radar information  
radar_info = pydarn.SuperDARNRadars.radars[6]  
print(radar_info.hardware_info.beams)
```

16



UNIVERSITY OF
SASKATCHEWAN





Tutorial: Documentation and Citing



Read the Docs

[PyDARN Documentation: Home](#)

pyDARN is DOI'd on Zenodo: [SuperDARN/pydarn: pyDARN v1.0.0 release](#)

Please see documentation for more information and how to reference SuperDARN.



UNIVERSITY OF
SASKATCHEWAN





Tutorial: Future Work

Bug fixes:

- RAWACF reading
- Missing radars in Hardware retrieval code
- Fixed logging file creation
- IQDAT borealis conversion

Plots:

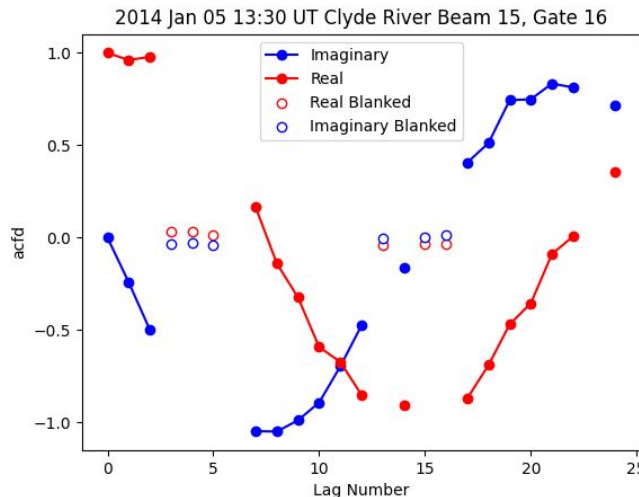
- ACF
- Fan Plots

IO:

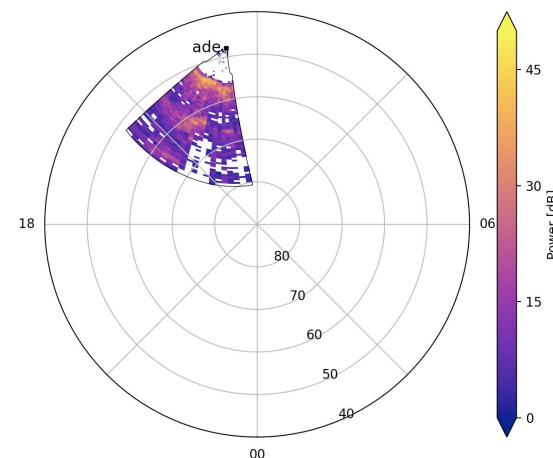
- New Borealis file version
- Data abstraction

pyDARN:

- Online through superdarn.ca
Coming Soon!



Adak Island SuperDARN Fan Plot - 2013/10/14 - 00:50



UNIVERSITY OF
SASKATCHEWAN





Tutorial: Issues

What to do when you run into a problem with pyDARN?

1. Go to **pyDARN** on GitHub: <https://github.com/superdarn/pydarn>
2. Select *issues*
3. Add a concise title of the problem or request
4. Add the following information:
 - a. Description of the problem
 - b. The code, OS, and python version you are using
 - c. The error, bug you found
5. Please add any labels and projects
6. Be patient and kind :)



UNIVERSITY OF
SASKATCHEWAN





How to Help: Testing

Installing a specific **pyDARN** branch: `$ pip3 install https://github.com/superdarn/pydarn@<branchname>`

If you want to change between branches you could install each branch in a different virtual environment or clone the repository:

```
$ git clone https://github.com/superdarn/pydarn
$ cd pydarn
$ git checkout develop
$ python3 setup.py install --user
```

To change branches:

```
$ git fetch
$ git checkout <branch name>
$ python3 setup.py install --user
```



UNIVERSITY OF
SASKATCHEWAN





How to Help: Code Reviews

Developers *should* provide test code in the Pull Request that you can try, as well you can review the code!

Who can code review in **pyDARN**?

- ANYONE!
- We need scientists for the mad visuals
- We need programmers for technical and style
- We need beginners to learn the ropes and ask the simple obvious questions

If you are testing or code reviewing and all the boxes are checked off, please merge the code. Developers of Pull Requests should merge their own code. You can always ask too.



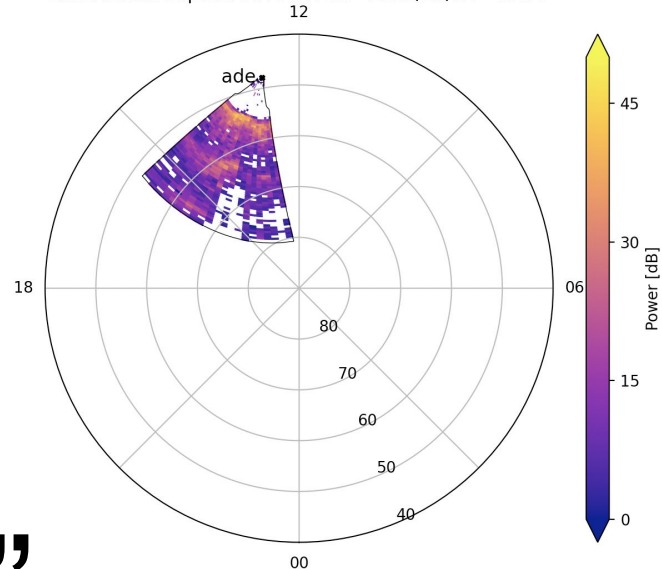
How to Help: Style Guide

You wanna be a developer:

- plan/design your code out
- Follow PEP 8 style guide
- Follow numpy/pandas documentation style guide
- Learn some markdown if you need to some documentation
- Remember to comment and use descriptive variable name
- Be ready for feedback in a collaborative setting!
- Don't be afraid to reach out and ask for help!

“ Fan plots were developed in [pyDARN](#) by a first time python user who uses IDL for his research ”

Adak Island SuperDARN Fan Plot - 2013/10/14 - 00:50



UNIVERSITY OF
SASKATCHEWAN





How to Help: Pull Request

Once you have developed your code, create a pull request and follow these steps:

1. Merge to **develop** or a sub-branch. *Not main (a.k.a master)*
2. Provide a concise title
3. Provide a detailed description with the following information:
 - a. Scope - what does this code do what does it not do
 - b. Reference any issues if there is one
 - c. Describe the idea behind the code
 - d. How does one test it?
 - e. What is the expect output?
 - f. Is there anything else needed to be done for the PR .i.e., Documentation, Unit tests
4. Add any useful labels
5. Add any projects related/milestone's
6. Submit!



UNIVERSITY OF
SASKATCHEWAN





How to Help: Becoming a PyDARNer

Get involved with a SuperDARN community project, learn from programmers and collaborate with the scientists. Any help guarantees you are an author on the DOI and great for resumes, or CV for community involvement in developing software.

Contact me at: marina.schmidt@usask.ca to get on the mailing list for **pyDARN** developer meetings and if you want to become a collaborator on **pyDARN**



UNIVERSITY OF
SASKATCHEWAN





Thank you

- Angeline Burrell -- for getting me involved
- SuperDARN community
- UNIS

