The Python Arm Radar Toolkit: The Expected and Unexpected Uses of an Open Source Radar Toolkit

Zach Sherman[1], Scott Collis[1], Robert Jackson[1], Mark Picel[1], and Jonathan Helmus[2]

[1]Argonne National Laboratory
[2]Continuum Analytics

American Meteorological Society's Eighth Symposium on Advances in Modeling and Analysis Using Python

The Python ARM Radar Toolkit: Py-ART

- The Python ARM Radar Toolkit, Py-ART, is an open source Python module containing a growing collection of weather radar algorithms and utilities build on top of the Scientific Python stack and distributed under the 3-Clause BSD license.

- Py-ART is used by the Atmospheric Radiation Measurement (ARM) Climate Research Facility and by others in the scientific community for working with data from a number of weather radars and instruments.

- When mentioning expected uses, ARM was working on a way or needed a tool to read, analyze and visualize the large amount of ARM radar data

- Py-ART is hosted on GitHub at http://arm-doe.github.io/pyart/

ARM CLIMATE RESEARCH FACILITY

Argonne NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY | Office of Science

The Python ARM Radar Toolkit: Py-ART

Py-ART can read a variety of file formats such as:

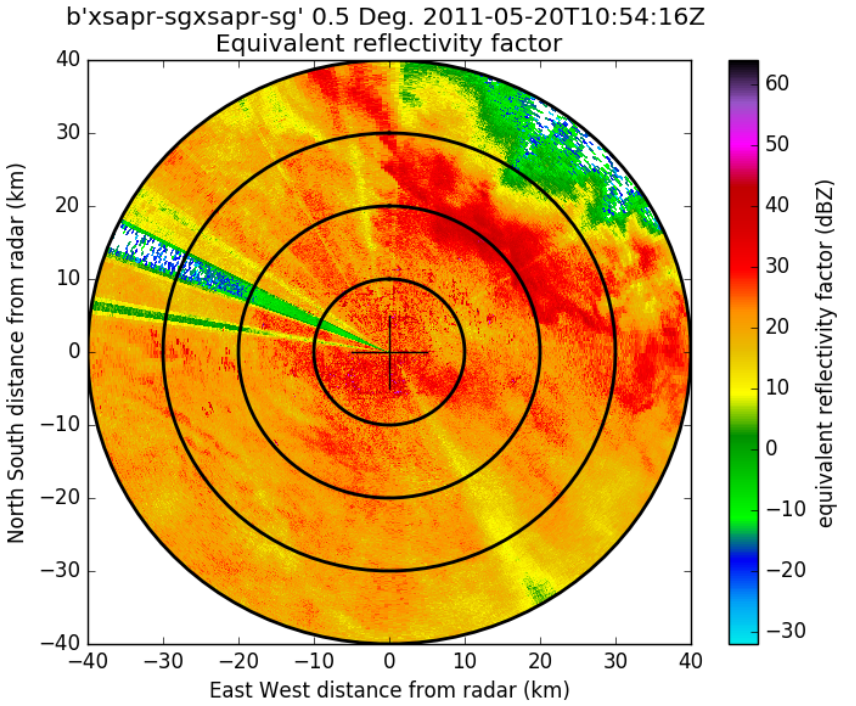
- NEXRAD level 2 and 3
- cfradial
- mdv
- sigmet
- rsl

Py-ART has expanded to now include the reading of file formats such as:

- rainbow
- odim h5
- radx
- sinarame h5
- gamic hdf5
- IPHEx NOXP

The Python ARM Radar Toolkit: Py-ART

```
print(__doc__)

# Author: Jonathan J. Helmus (jhelmus@anl.gov)
# License: BSD 3 clause

import matplotlib.pyplot as plt
import pyart

filename = 'XSW110520105408.RAW7HHF'

# create the plot using RadarDisplay (recommended method)
radar = pyart.io.read_rsl(filename)
display = pyart.graph.RadarDisplay(radar)
fig = plt.figure()
ax = fig.add_subplot(111)
display.plot('reflectivity', 0, vmin=-32, vmax=64.)
display.plot_range_rings([10, 20, 30, 40])
display.plot_cross_hair(5.)
plt.show()
```



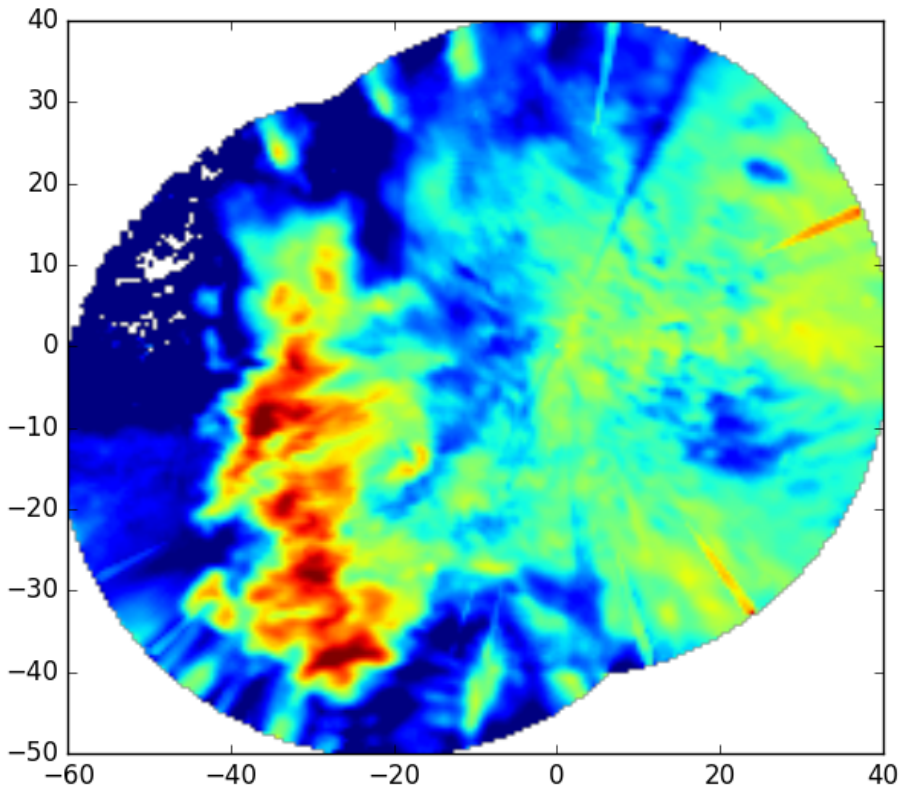
ARM CLIMATE RESEARCH FACILITY

Argonne NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY | Office of Science

The Python ARM Radar Toolkit: Py-ART

```
print(__doc__)

# Author: Jonathan J. Helmus (jhelmus@anl.gov)
# License: BSD 3 clause

import matplotlib.pyplot as plt
import pyart

# read in the data from both XSAPR radars
XSAPR_SW_FILE = 'swx_20120520_0641.nc'
XSAPR_SE_FILE = 'sex_20120520_0641.nc'
radar_sw = pyart.io.read_cfradial(XSAPR_SW_FILE)
radar_se = pyart.io.read_cfradial(XSAPR_SE_FILE)

# filter out gates with reflectivity > 100 from both radars
gatefilter_se = pyart.filters.GateFilter(radar_se)
gatefilter_se.exclude_transition()
gatefilter_se.exclude_above('corrected_reflectivity_horizontal', 100)
gatefilter_sw = pyart.filters.GateFilter(radar_sw)
gatefilter_sw.exclude_transition()
gatefilter_sw.exclude_above('corrected_reflectivity_horizontal', 100)

# perform Cartesian mapping, limit to the reflectivity field.
grid = pyart.map.grid_from_radars(
    (radar_se, radar_sw), gatefilters=(gatefilter_se, gatefilter_sw),
    grid_shape=(1, 201, 201),
    grid_limits=((1000, 1000), (-50000, 40000), (-60000, 40000)),
    grid_origin = (36.57861, -97.363611),
    fields=['corrected_reflectivity_horizontal'])

# create the plot
fig = plt.figure()
ax = fig.add_subplot(111)
ax.imshow(grid.fields['corrected_reflectivity_horizontal']['data'][0],
          origin='lower', extent=(-60, 40, -50, 40), vmin=0, vmax=48)
plt.show()
```



ARM CLIMATE RESEARCH FACILITY

Argonne NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY | Office of Science

The Python ARM Radar Toolkit: Py-ART

Corrections

- Velocity Dealiasing
- Attenuation calculation
- Phase processing
- Despeckle
- RhoHV noise correction
- Radar data bias correction

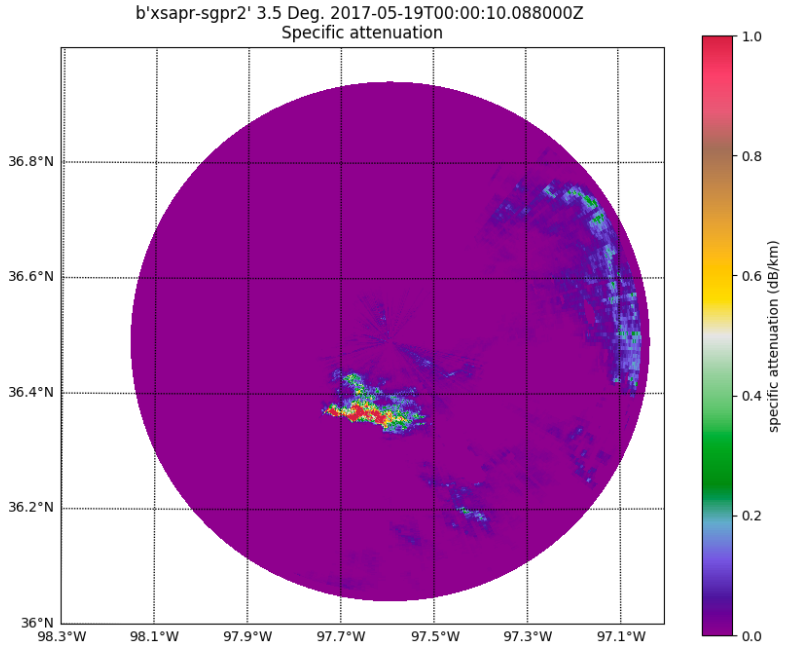
Gate filters can also be applied depending a variety of circumstances and choices.

Retrievals

- Calculate signal to noise ratio from reflectivity field and noise in dBZ
- Estimate rain rate from specific attenuation
- Calculate texture of the differential phase field
- Calculate grid displacement using phase correlation
- And more!

The Python ARM Radar Toolkit: Py-ART

```
# Calculating attenuation by using pyart.
spec_at, cor_z_atten = pyart.correct.calculate_attenuation(
    radar, 0, refl_field='reflectivity',
    ncp_field='normalized_coherent_power',
    rhv_field='cross_correlation_ratio',
    phidp_field='filtered_corrected_differential_phase',
    a_coef=attenuation_a_coef)
```



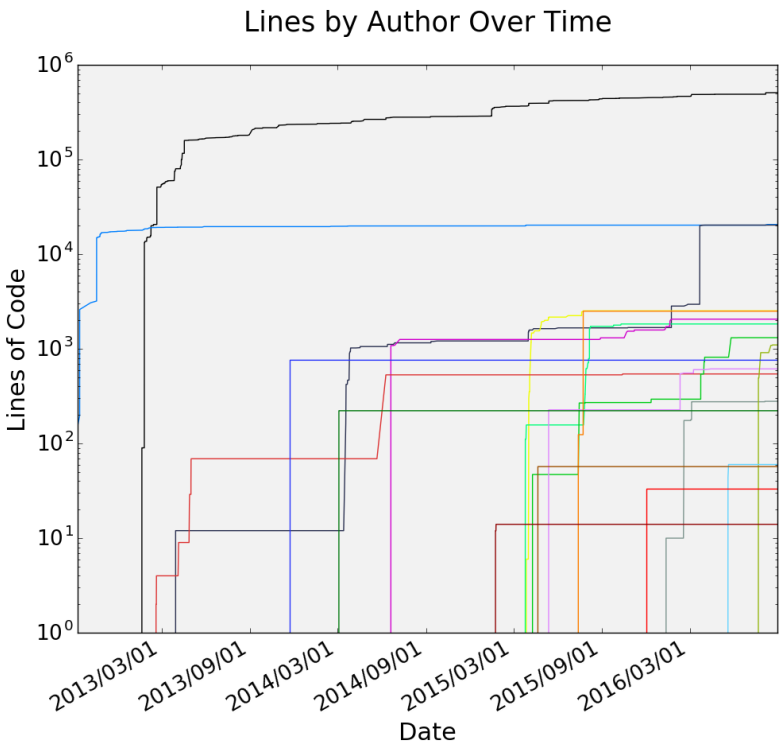
Engineered for New Contributors

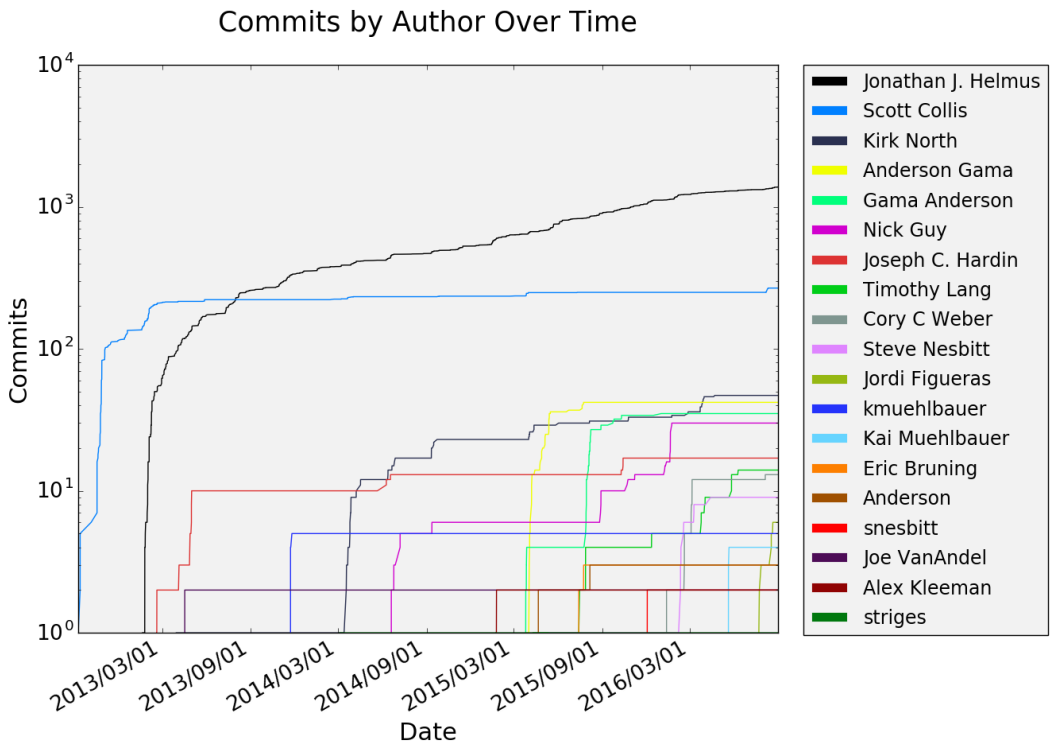
- Py-ART was carefully engineered using continuous integration for automatic testing, and to provide a friendlier user environment with ways for users to contribute their own packages.
- Continuous integration allows for the testing of new and existing code. This helps prevent new code from breaking the preexisting code.
- These aspects have paid off, because over time, Py-ART has benefited from numerous and diverse contributions from users around the globe.

Travis CI

AppVeyor

Engineered for New Contributors





ARM CLIMATE RESEARCH FACILITY

U.S. DEPARTMENT OF ENERGY | Office of Science

Py-ART the Unexpected Uses

Py-ART and MeteoSwiss

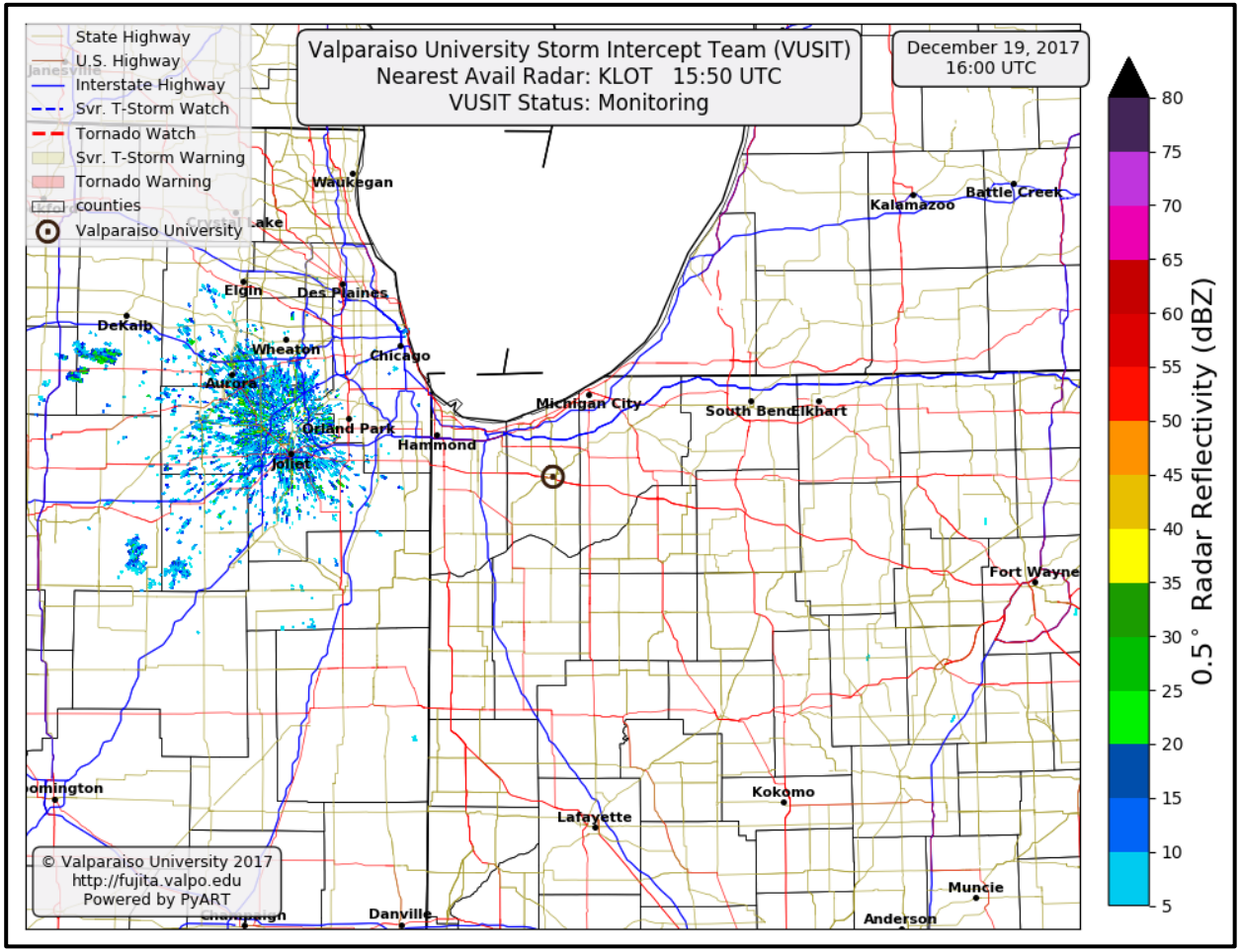
- Jordi Figueras i Ventura and members from MeteoSwiss, officially the Federal Office of Meteorology and Climatology, has been using Py-ART in their program Python Radar Processing (Pyrad). https://github.com/meteoswiss-mdr/pyrad

- Not only are they using Py-ART in their program Pyrad, MeteoSwiss also has a Py-ART branch with many new features and improvements. https://github.com/meteoswiss-mdr/pyart/tree/pyart-mch

- The team at MeteoSwiss plans on creating pull requests, depending on discussion in FY18, on which functions to add to the master branch at ARM-DOE on GitHub.

Real time Radar at Valparaiso University

- Valparaiso University Meteorology Program uses Py-ART for real time radar scans from the KLOT NEXRAD radar on their Online Weather Center. http://fujita.valpo.edu/radar/reflectivity/ref



ARM CLIMATE RESEARCH FACILITY

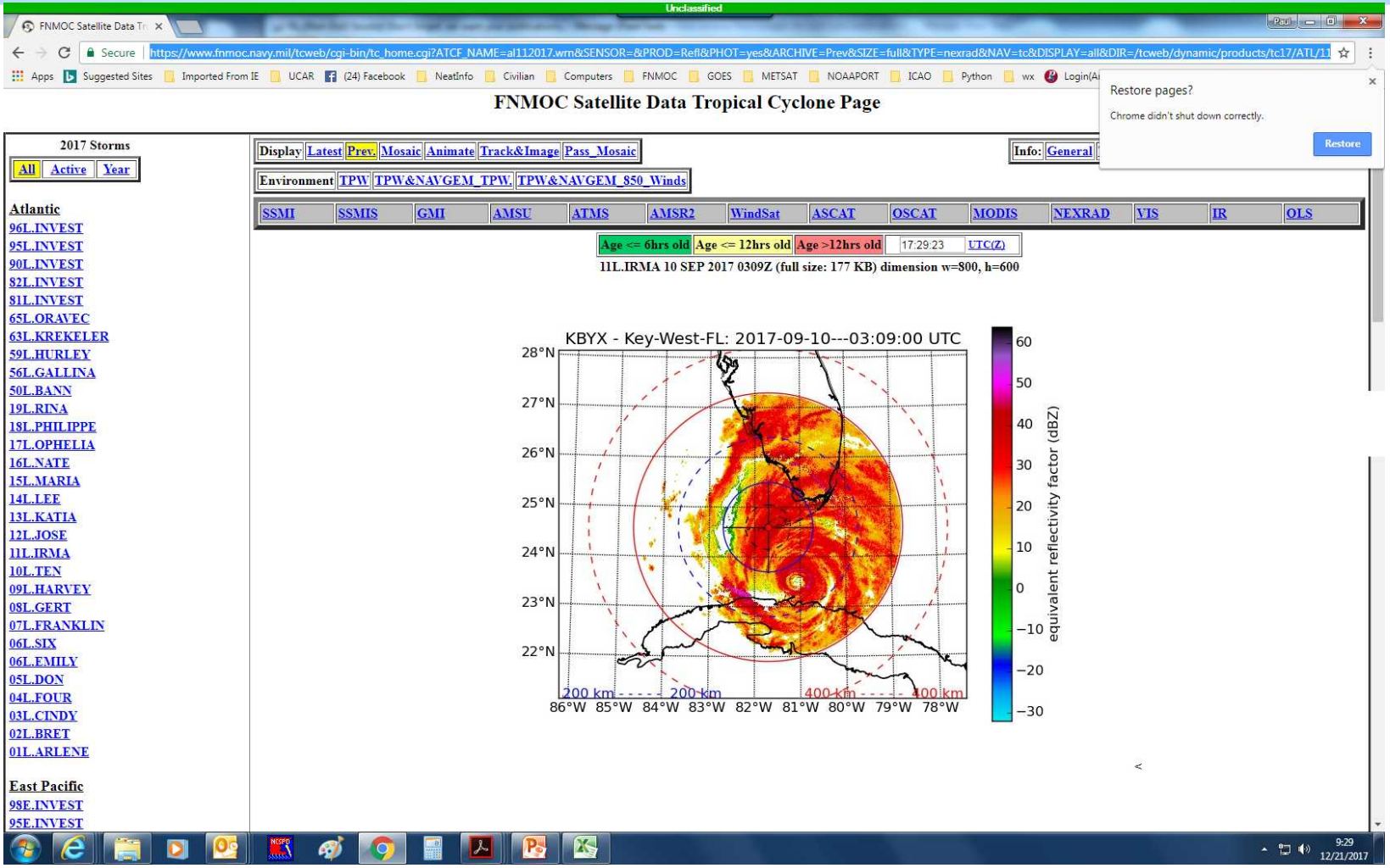
Argonne NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY | Office of Science

The NAVY

- Uses Py-ART to read NEXRAD level 2 data.
- Trying to include the UF format sites, because Py-ART has the capability to read UF format.
- Mostly process coastal radar sites such as the Gulf coast, East coast, Hawaii, Guam, Kadena, and Korea, areas that are susceptible to hurricanes/typhoons.
- Some other radar sites and branches that use Py-ART are MARINE CORPS, USAF and NAVY non-NEXRAD radar sites.

The NAVY

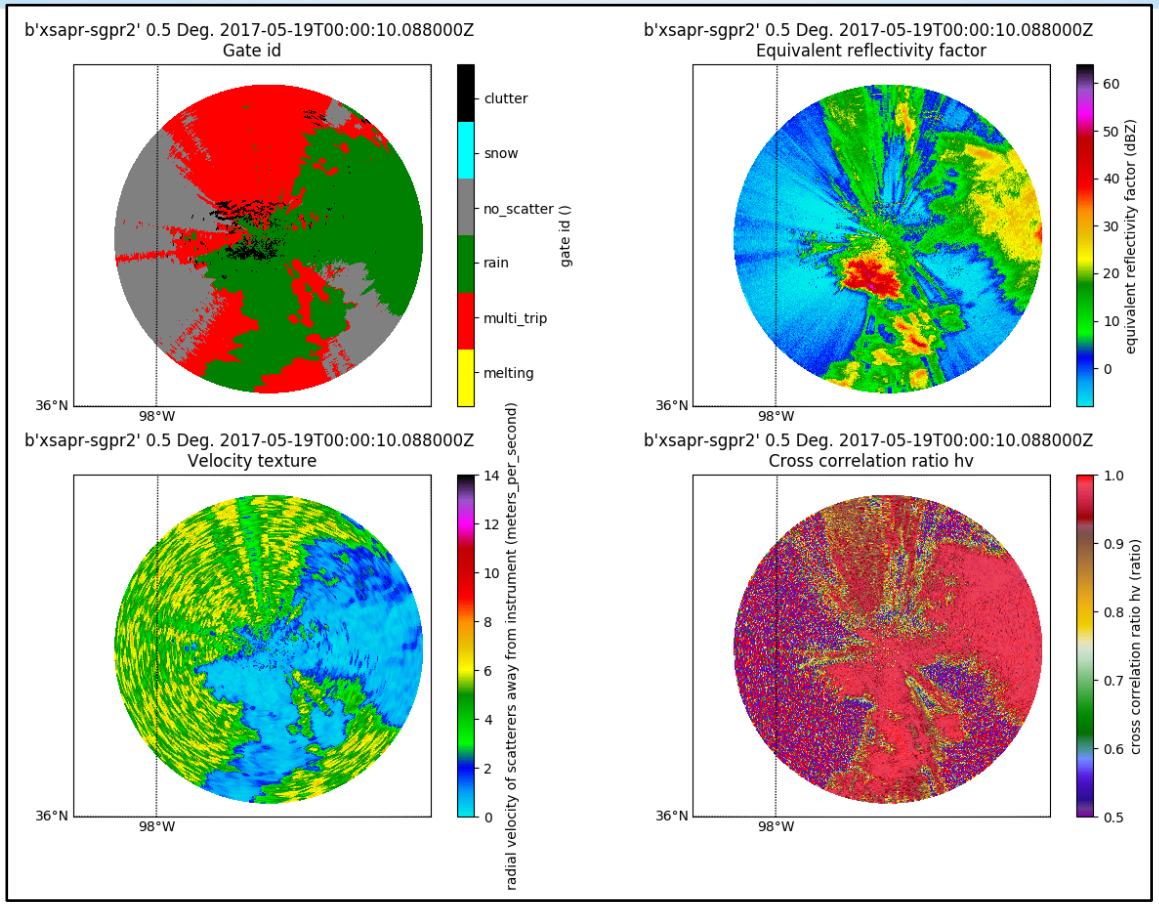


ARM CLIMATE RESEARCH FACILITY

Argonne NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY | Office of Science

Using Scikit-fuzzy and Py-ART for Hydrometeor Classifications

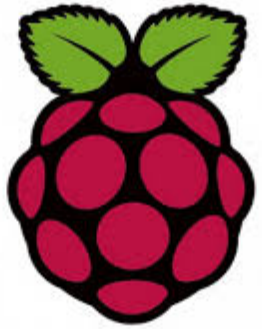


Analyzing Large Radar Datasets Using Python. Robert Jackson, ANL, Argonne IL; S. Collis, Z. Sherman, G. Palanisamy, S. Giangrande, J. Kumar, J. C. Hardin

ARM CLIMATE RESEARCH FACILITY
Argonne NATIONAL LABORATORY
U.S. DEPARTMENT OF ENERGY | Office of Science

The Raspberry Pi

- BerryConda a package created by Jonathan Helmus, allows for the installation of Anaconda and a variety of packages to be installed onto a Raspberry Pi.
- Scott Collis was able to install Anaconda on a Raspberry Pi and was able to install Py-ART.
- This allows for a cheap and easy distribution of open-source software to not only the radar sites, but can also be used for educational purposes.



The Raspberry Pi

```
freetype-2.7-0 100% |################################################################################
pyparsing-2.2. 100% |################################################################################
pytz-2017.2-py 100% |################################################################################
cycler-0.10.0- 100% |################################################################################
matplotlib-2.0 100% |################################################################################
(pirad) pi@thunder:~ $ python
Python 3.6.2 | packaged by rpi | (default, Jul 19 2017, 12:58:40)
[GCC 4.9.2] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import pyart

## You are using the Python ARM Radar Toolkit (Py-ART), an open source
## library for working with weather radar data. Py-ART is partly
## supported by the U.S. Department of Energy as part of the Atmospheric
## Radiation Measurement (ARM) Climate Research Facility, an Office of
## Science user facility.
##
## If you use this software to prepare a publication, please cite:
##
##     JJ Helmus and SM Collis, JORS 2016, doi: 10.5334/jors.119

/home/pi/berryconda3/envs/pirad/lib/python3.6/site-packages/pyart/config.py:22: DeprecationWarnin
b; see the module's documentation for alternative uses
  import imp
>>> radar = pyart.io.read('/home/pi/KLOT20150610_124714_V06')
>>> print(radar.name)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'Radar' object has no attribute 'name'
>>> print(radar.fields.keys())
dict_keys(['reflectivity', 'differential_phase', 'velocity', 'differential_reflectivity', 'spectru
>>> print(radar.fields['reflectivity']['data'].mean())
6.88169196934
>>>
```

ARM CLIMATE RESEARCH FACILITY

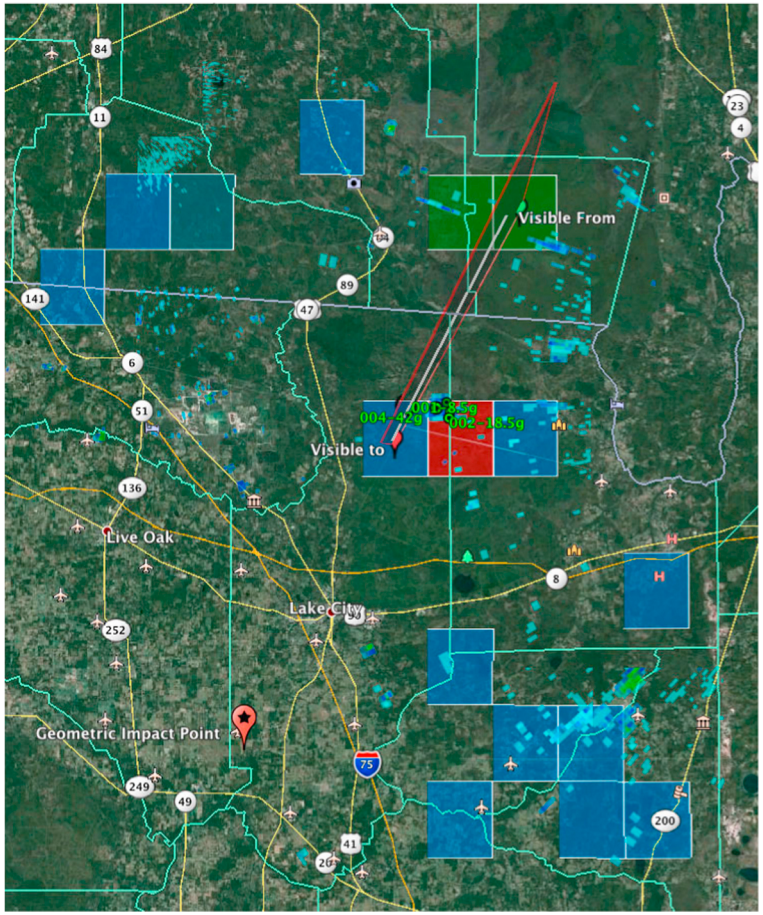
Argonne NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY | Office of Science

Meteorite Strewn Fields

- Originally, to attempt to determine a meteorite strewn field, researchers would have to go off of reported sightings and would have to go to multiple NEXRAD sites and try to catch signs of the meteorite fall on radar scans Hankey et al., (2017).
- The automation of detecting strewn fields.
- NEXRAD data is available on Amazon Web Services and can be downloaded using the python package boto. Py-ART can read, analyze and plot the downloaded data.
- Additionally it is believed that due to the Py-ART's added support for reading Rainbow radar files, it is possible to extend radar meteorite recovery capabilities to the international community due to the wide use of Rainbow files in Europe and other locations throughout the globe Hankey et al., (2017).

Meteorite Strewn Fields

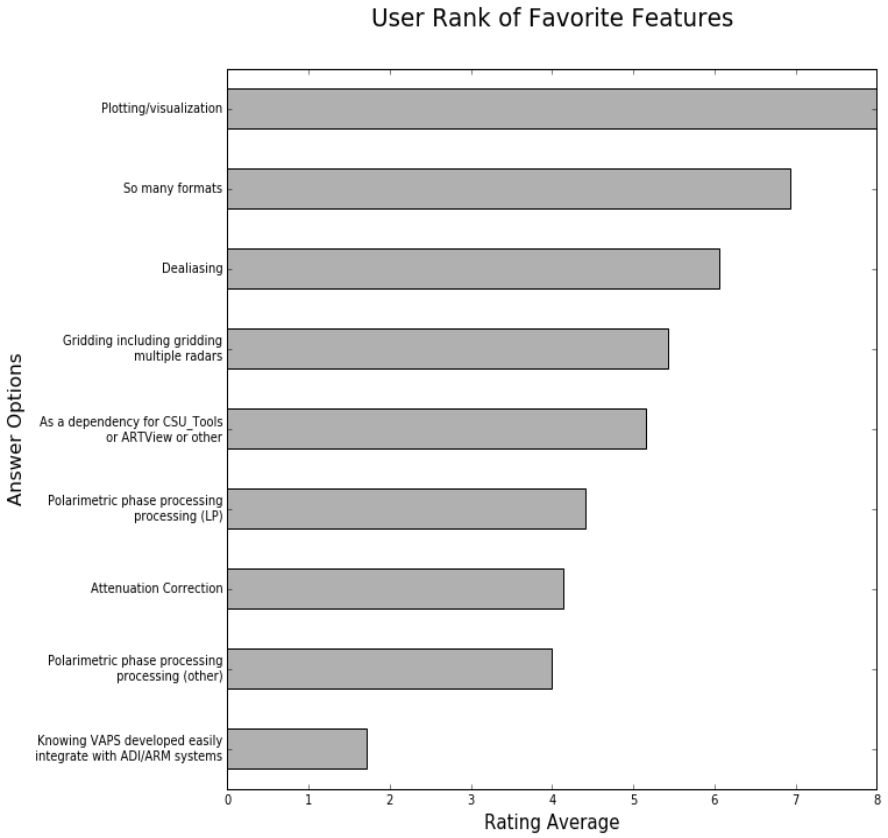


Hankey et al., (2017).


AHEAD
EXPECT THE
UNEXPECTED

Py-ART Roadmap

- The Py-ART roadmap is a document for development for Py-ART.
- It sets priorities for ARM funded work and it advises the community which contributions are desired and that ARM software engineers will work on.
- Involved surveys, directly talking to the community and ARM.



Conclusion

- Py-ART was originally created as an ARM Tool.
- Due to being open source and being engineered with continuous integration the amount of contributors and users has been increasing greatly.
- Due to the size of the community and the added abilities of Py-ART, it uses expanded outside the scope of ARM.
- With more contributors and users, the capabilities of Py-ART will continue to grow.
- As Py-ART continues to grow there may be many new unexpected uses.
- The Py-ART roadmap will also help guide ARM engineers in future development.

Packages That Use Py-ART

- ARTView https://github.com/nguy/artview
- SIngleDop https://github.com/nasa/SingleDop
- PyTDA https://github.com/nasa/PyTDA
- DualPol https://github.com/nasa/DualPol
- CSU Radar Tools https://github.com/CSU-Radarmet/CSU_RadarTools
- MultiDop https://github.com/nasa/MultiDop

References

- Hankey, M. et al., (2017). AMSNEXRAD-Automated detection of meteorite strewnfields in doppler weather radar, In Planetary and Space Science, Volume 143, Pages 199-202, ISSN 0032-0633, DOI: https://doi.org/10.1016/j.pss.2017.02.008

- Helmus, J.J. & Collis, S.M., (2016). The Python ARM Radar Toolkit (Py-ART), a Library for Working with Weather Radar Data in the Python Programming Language. Journal of Open Research Software. 4(1), p.e25. DOI: http://doi.org/10.5334/jors.119

Thank you for Listening!

Questions?

Contact Information: zsherman@anl.gov