ACT for Science

Zachary Sherman[1], Kenneth Kehoe[2], Maxwell Grover[1], Matthew Tuftedal[1], Joseph O'Brien[1], Bobby Jackson[1], Adam Theisen[1], Maxwell Levin[3], Michael Giansiracusa[4], Jenni Kyrouac[1], Alyssa Sockol[2], Corey Godine[2] and Austin King[2]
1 – Argonne National Laboratory, 2 – CIWRO, University of Oklahoma,
3 – Pacific Northwest National Laboratory, 4– Oak Ridge National Laboratory

What is ACT?

The Atmospheric data Community Toolkit (ACT) is an open-source Python library for exploring and analyzing time-series-based atmospheric research datasets. ACT offers modules for the entire scientific lifecycle to aid researchers in their efforts, while also being stable enough for use in ARM infrastructure. ACT has been supported by ARM since 2019. ACT continues to expand, welcoming and encouraging contributions from its users and the broader scientific community.

Connecting Valuable Datasets

ACT offers a variety of discovery modules to easily download data from ARM and other organizations that can aid in your research

- ARM Live Data Web Service (Fig. 1)
- NOAA Physical Sciences Laboratory (Fig. 1)
- NOAA ESRL Global Monitoring Laboratory
- EPA AirNow Surface Observations (Fig. 2)
- Automated Surface Observing Systems (IEM API)
- USDA CropScape Yearly Crop Type information

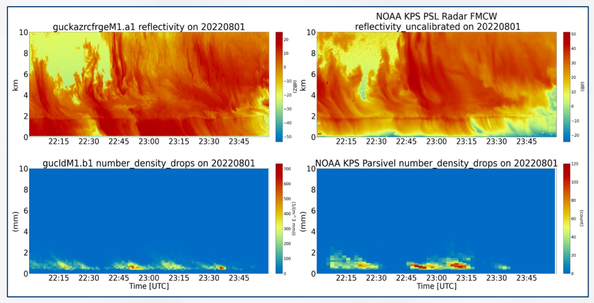

Figure 1. Plot of ARM KAZR and Parsivel (left top and bottom) and the NOAA Kettle Ponds FMCW and Parsivel (right top and bottom).

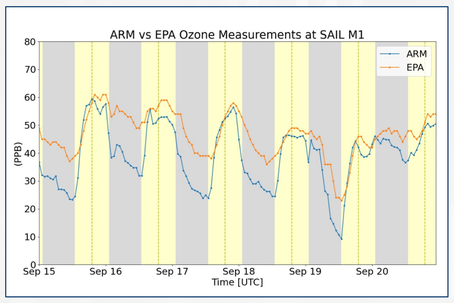

Figure 2. Comparison of ARM (S2) and EPA (Gothic) ozone concentrations at SAIL.

Opening Retrievals

There are many benefits to contributing retrievals (or other codes) to ACT including transparency, flexibility, broader community use and feedback, reduced duplication of effort, and the inclusion of the contributor(s) on the ACT DOI and authorship list. We ask that ARM/ASR researchers consider contributing code to ACT as part of their research workflows to benefit the broader community. A few examples of the retrievals already available in ACT include:

- Sea Surface Temperature from the IRTSST
- Scattering and BC Mass Size Distributions from the SP2 (PySP2)
- Wind Profile Retrieval from Doppler Lidars (Fig. 4).
- Sky Temperatures from the AERI (Fig. 5)
- Radiosonde Derived PBL Heights
- Radar/Lidar Cloud Base Heights

Where available, the citation for the reference paper(s) or technical report(s) for each retrieval is referenced in the code and is available through the ACT API Reference Manual to help drive users to cite the corresponding reference materials.

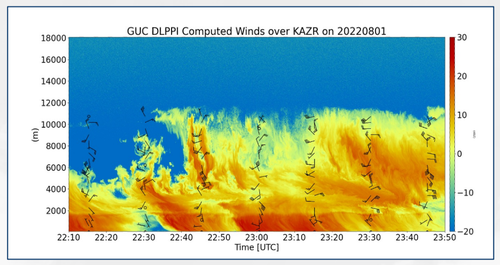

Figure 4. ARM KAZR reflectivity with winds calculated from the ARM DL during the SAIL campaign.

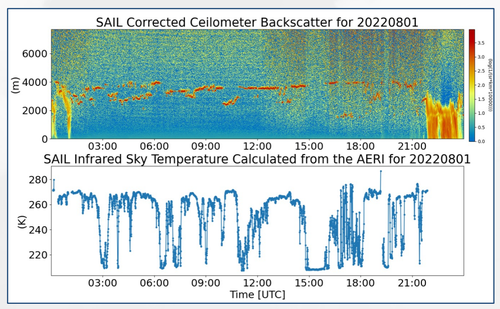

Figure 5. Plot of ARM ceilometer backscatter and infrared sky temperature calculated from the AERI during the SAIL campaign.

Standardizing Datasets

ACT has the capability to read a variety of datasets with specific readers for ARM, NOAA GML, NOAA PSL, raw MPL, ICARTT, csv, and more! Data are read into the Xarray Dataset which ACT tools are based around. ACT and Xarray make it easy to work with other python software as well and contribute to building a software ecosystem (Fig. 3). Additionally, there is functionality for PIs to create NetCDF files that mimic ARM's files (variables, metadata, etc.) to standardize PI contributions..

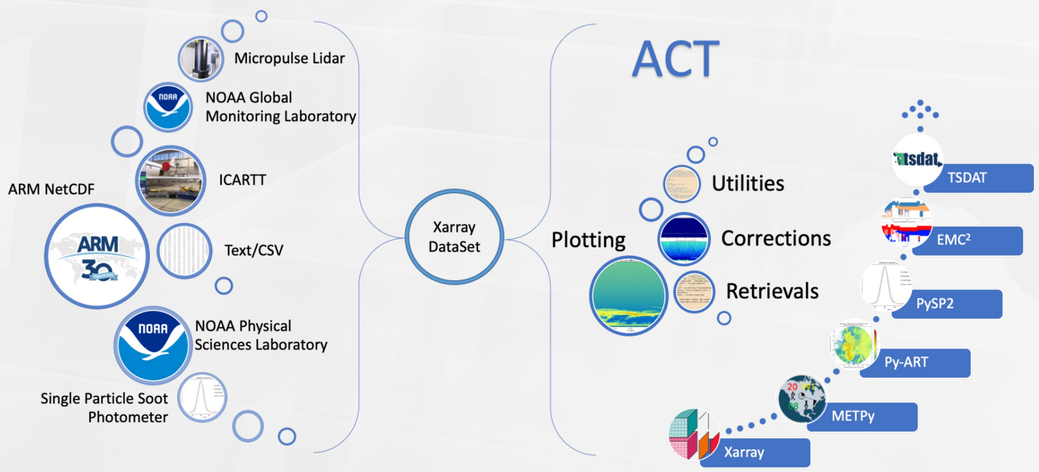

Figure 3. The ACT ecosystem. ACT serves to standardize various inputs (left) into a consistent Xarray Dataset (middle) that is used throughout ACT (right). This Xarray dataset and/or ACT are used in a variety of other python libraries that enable easier sharing of codes and resources. Far right shows packages that can be utilized when using ACT and Xarray.

Sharing Code In Lieu of Data

ACT streamlines the process of downloading data, performing calculations, retrievals, corrections, and saving the results to NetCDF files, making it much easier to share reproducible analyses via a single Python script (Fig. 6). ACT's open-source tools can also allow organizations like ARM to process data on-demand when users order data (Fig. 7).

```python
import act
import matplotlib.pyplot as plt


# Use arm username and token to retrieve files.
token = 'arm_token'
username = 'arm_username'

#Specify datastream and date range for KAZR data
ds_kazr = 'guckazrcfrgeM1.a1'
startdate = '2022-08-01'
enddate = '2022-08-01'

# Download and read in the ARM KAZR data
act.discovery.download_data(username, token, ds_kazr, startdate, enddate)
kazr_files = glob.glob(''.join(['./',ds_kazr,'/*nc']))
kazr = act.io.armfiles.read_netcdf(kazr_files[-2:])

# Download and read in the corresponding doppler lidar data
ds_dl = 'gucdlppiM1.b1'
act.discovery.download_data(username, token, ds_dl, startdate, enddate)
dl_ppi_files = glob.glob(''.join(['./',ds_dl,'/*cdf']))
dl_ppi = act.io.armfiles.read_netcdf(dl_ppi_files[-9:])

# Calculate the winds from the gucdlppi dataset.
wind_obj = act.retrievals.compute_winds_from_ppi(dl_ppi, remove_all_missing=True, snr_threshold=0.002)

# Create a display object and plot out the KAZR reflectivity with Wind Bards Overlaid
display = act.plotting.TimeSeriesDisplay({"GUC DLPPI Computed Winds over KAZR": wind_obj,
                                          "guckazrcfrgeM1.a1": kazr,}, figsize=(20, 10))
display.plot('reflectivity', dsname='guckazrcfrgeM1.a1', cmap='act_HomeyerRainbow', vmin=-20, vmax=30)
display.plot_barbs_from_spd_dir('wind_speed', 'wind_direction', dsname='GUC DLPPI Computed Winds over KAZR',
                                invert_y_axis=False)
```

SCAN ME — ACT Basics Tutorial

SCAN ME — ACT Blog Post on SAIL Data

Figure 6. Example script of how simple it is to share code for downloading, processing, applying retrievals, and visualizing data. Top QR codes link to the ACT Basics Tutorial from the Open Science Workshop (Left) and a blog post on exploring SAIL data from ARM and NOAA using ACT (right).

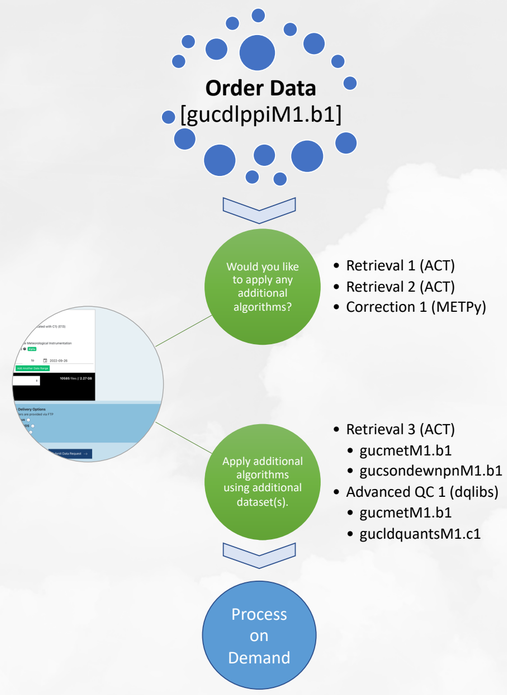

Figure 7. Workflow of how open-source tools could be implemented into data portals to enable processing-on-demand for advanced retrievals.

https://github.com/ARM-DOE/ACT

U.S. DEPARTMENT OF ENERGY | Office of Science