Atmospheric data Community Toolkit (ACT) Roadmap

Introduction and Goal

The Atmospheric data Community Toolkit (ACT) is an Atmospheric Radiation Measurement User Facility (ARM) supported python library for working with atmospheric time series x N-dimension datasets. ACT has functions for every aspect of the scientific process. Current modules include:

- Discovery – Access APIs for accessing data
- IO – Read/Write data
- Plotting – Easy visualization of data
- QC – Working with existing QC and applying new tests
- Corrections – Apply known corrections necessary for using data
- Retrievals – In depth calculations that aren’t suited for their own repo
- Utils – General calculations/utilities not suited for the other modules

ACT aims to reduce the redundant coding performed by individual researchers and focus that effort into developing capabilities that can be utilized by the broader community. This document is a guiding roadmap for development of ACT.

Atmospheric data Community Toolkit

ACT grew out of a need to easily work with atmospheric data across sources (Argonne, Array of Things, Atmospheric Measurement User Facility (ARM), Ameriflux, etc) and was initially supported by Argonne National Laboratory through a laboratory directed research and development activity. Knowledge gained from the development and maintenance of the Python ARM Radar Toolkit (Py-ART) (Helmus and Collis 2016) was leveraged to ensure a solid framework was developed on which to advance the toolkit. ACT is currently being supported and maintained by ARM.

The initial development focused on efforts to read data from a variety of different organizations and ensure that there's a broad assortment of scripts to visualize that data. Collaborations began very early in the ACT development and allowed ACT to immediately begin serving the community.

At the time of writing, ACT is on Version 0.4.3 and has variety of features that are documented in the google doc linked below. It is available for installation through conda-forge and pip under the name act-atmos. Zenodo has been incorporated into the repository for automatic DOI generation for each release. Coveralls has also been implemented in the testing to ensure that any additions do not substantially decrease the unit test coverage.
https://docs.google.com/document/d/15moDtl5gejPPnWIVgTRspNcKlenwwZv_3TB0QU6Gqds/edit?usp=sharing

Value of ACT to the Community

The avenue of fostering collaboration and sharing of code will be the largest benefit of ACT to the community. ACT will allow individual groups to break out of their silo development efforts and connect with development from other groups. It is then that the science and infrastructure communities can start to remove redundant efforts and effectively advance, building off of one another as opposed to recreating all the building blocks from the ground up for their effort.

Overarching Goals for Next Six to Twelve Months

This roadmap will be a working strategy to advance ACT. Due to the rapid increase in content in ACT, we are updating the roadmap every six to twelve months to ensure it accurately reflects the current priorities. There are several key areas that ACT will look to continue development and advance in over the upcoming years.

Unit Test Coverage

The implementation of coveralls to track the unit test coverage highlighted the need to increase said coverage. Currently, the coverage is 74% and the goal will be to bring this level up to 90% or better in the next 12 months.

Discovery

There is a need to expand the toolkit for working with other agencies data portals and their APIs.

I/O

Instrument vendors supply their data in a variety of different formats. ACT will look to build on the I/O capabilities to expand the use of ACT out to different data types. Recently, a binary reader for micropulse lidar data using mpl2nc was added.

Quality Control

There are various ways the different programs handle quality control (QC). There is a need to easily incorporate quality control information into ACT from these varying sources. This would include masking of data based on QC data both internal to the dataset and external through API access such as the ARM data quality report web service. Addition of new quality control tests will also be a focus.

Retrievals

A goal of ACT is to bridge the science and infrastructure communities. In order to do that, developers will work with the science community to develop retrievals in python or processing functions on an as needed basis.

Education and Outreach

In order to build up a user base, effort will need to be put into developing tutorials and examples as well as working directly with users on questions and issues that arise.

Proposed Governance Structure

There is a need to ensure that this effort is responsive to the needs of stakeholders that rely on ACT. The proposed governance structure needs to be flexible and have the ability to expand as the capabilities of the toolkit expand. The roles required are:

Science Lead: Provides high level leadership for the toolkit, organizes outreach and education, and coordinates contributor and stakeholder input to form a long-term vision for the project. The science lead will also coordinate reviews of the science behind a pull request to ensure accuracy to the literature. The science lead will make a judgement on if a pull request requires more review or can be accepted as is.

Lead Developer: Responsible for overall architecture of the project. Final arbiter in what pull requests to accept. Develops the required style guidelines and coordinates the associate developers. Coordinates contributions from associate developers to a Contributors Guide.

Associate Developer: Responsible, as time allows, for doing an initial check of pull requests for suitability and adherence to the Contributors Guide. Associate developers should come from a diverse background to ensure there is no single point of failure in providing support to ACT. The number and areas of expertise of the associate developers will need to increase and adapt as ACT grows.

Measuring Impact

The impact of ACT can be measured in multiple ways

1) **Growing the number of users and installs.** Success would mean growing the traffic from the period of the first roadmap (Fig. 1.) from over 100 views, 11 unique visitors, and 12 unique cloners, with emphasis on the latter. GitHub only provides the last 14 days' worth of statistics and in order to keep a detailed record over time, we would need to integrate an external tracking tool such as Google Analytics, or create a database and pull information from GitHub through its' API.
    a. The current statistics show 389 views, 23 unique visitors, and 40 unique cloners during the weeks 8/15-8/18/2020.

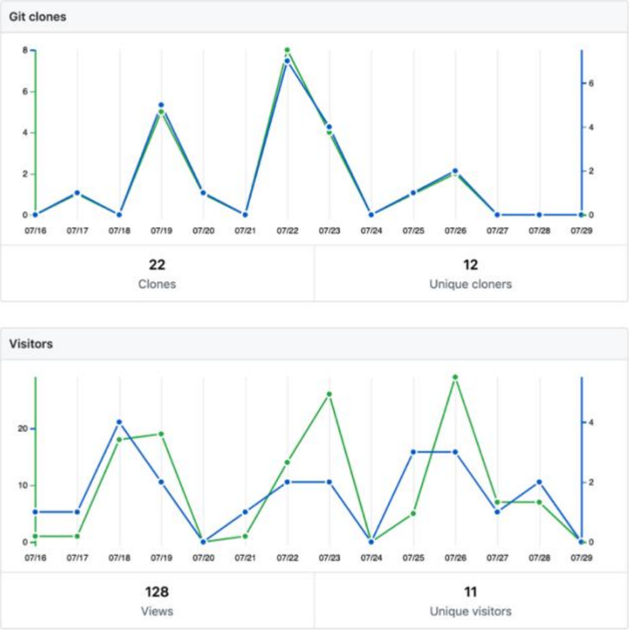

Figure 1. Github traffic from the week preceding July 29th, 2019.

2) **Number and success of dependent projects.** In order to be sustainable, ACT needs to have a clearly defined scope and anything that falls outside this scope would be better suited as a separate repository. Success can be measured by an increase to the number of dependents that are using ACT.
    a. There are currently 9 repositories that use ACT as a dependency.
3) **Papers and presentations using ACT.** Publications are treated by many user facilities as a metric of scientific impact. We will encourage users to cite the metadata paper that will be forthcoming but also the ACT and any integrated software DOIs.
    a. Cited by 1 paper that is currently in review.

The goal over the next few years will be to grow the user base, expand the capabilities of ACT, build a rich ecosystem of applications that rely on ACT and become a useful resource for the scientific community.

References

Giansiracusa, M., (2019). Python scripts for ARMLIVE Web service. Github: https://code.ornl.gov/ofg/armlive_getfiles

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