

Atmospheric data Community Toolkit (ACT) Roadmap

INTRODUCTION AND GOAL	1
ATMOSPHERIC DATA COMMUNITY TOOLKIT.....	1
VALUE OF ACT TO THE COMMUNITY	2
OVERARCHING GOALS FOR NEXT THREE YEARS.....	2
DISCOVERY	2
VISUALIZATION.....	2
QUALITY CONTROL.....	2
GENERAL UTILITIES	2
<i>Gridding</i>	2
<i>Program Specific Utilities</i>	3
RETRIEVALS	3
EDUCATION AND OUTREACH	3
PROPOSED GOVERNANCE STRUCTURE.....	3
MEASURING IMPACT	4
REFERENCES.....	5

Introduction and Goal

The Atmospheric data Community Toolkit (ACT) is a software package for easily reading, processing, and visualizing atmospheric data with varying number of dimensions. It 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 Program, Ameriflux, etc) and was supported by Argonne National Laboratory through a laboratory directed research and development activity (LDRD). 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.

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.

ACT leverages the ARM Live Web Service script, developed by Oak Ridge National Laboratory, for accessing ARM data (Giansiracusa 2019). The ARM Data Quality (DQ) Office in the Cooperative Institute for Mesoscale Meteorological Studies (CIMMS) at the University of Oklahoma (OU) have been contributing to the development of ACT as they transition their plotting code to Python. ACT serves as an avenue to get a wealth of ARM DQ Office functions and scripts available to the broader community.

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 Three Years

This roadmap will be a working strategy to advance ACT. There are a number of key areas that ACT will look to continue development and advance in over the upcoming years.

Discovery

There is a need to expand the toolkit for working with other agencies data portals and their APIs. For example, a new script to access crop type information from the USDA CropScape tool was recently added into the ACT toolkit which will be a big benefit to science using flux measurements.

Visualization

While there is a large number of scripts for visualizing data already, there will always be new capabilities that are needed in visualizing data. There will also be continuing efforts to improve and speed up existing plotting capabilities.

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. Easy visualization of the QC information will help users have a better understanding of the QC information and how it impacts their datasets

General Utilities

Gridding

While xarray does have inherent gridding capabilities, they may not be easy for new users to implement. Features for gridding up different instruments onto the

same time grid is needed and could be extended to help with the model to observation comparisons.

Program Specific Utilities

ARM PI's routinely submit data to ARM to be stored and disseminated from the IOP or PI data areas. The development of a data validator to ensure that the data products meet ARM/CF standards could help improve the general ease with which other users can work with the data.

ARM has a web service in which the DODs could be retrieved for current ARM instruments. An ACT compatible object could be prebuilt so that a user only needs to populate the data and some of the metadata. This object could then be easily written to netcdf in an ARM standard format. A generic PI DOD could always be created to help build PI datasets as well.

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 current traffic (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.

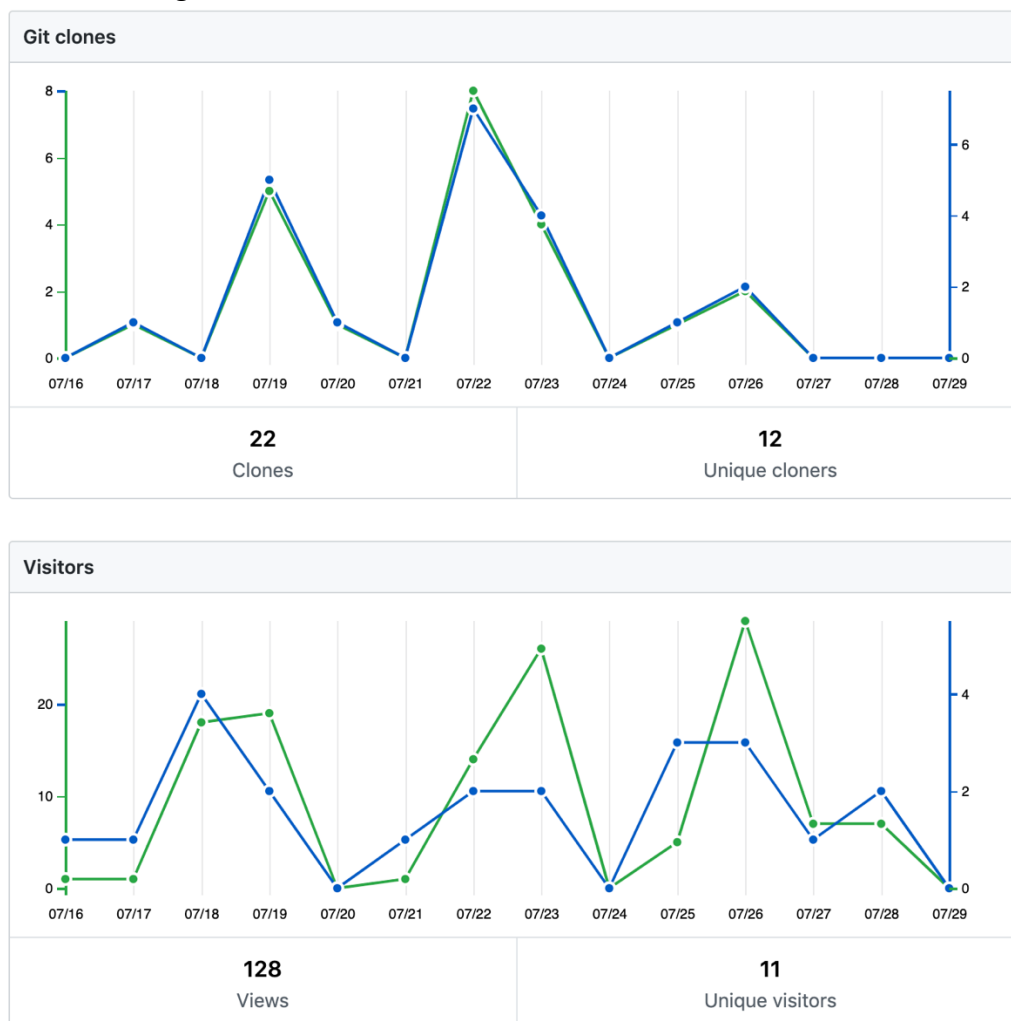


Figure 1. Github traffic from the week preceding July 29th, 2019.

- 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.

- 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.

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>