

Atmospheric data Community Toolkit (ACT)

Making the simple things simple

*K. Kehoe, A. Theisen, B. Jackson, Z. Sherman, A. Sockol, C. Godine, J. Kyrouac,
J. Hemedinger, S. Collis and M. Giansiracusa*

Reading Data



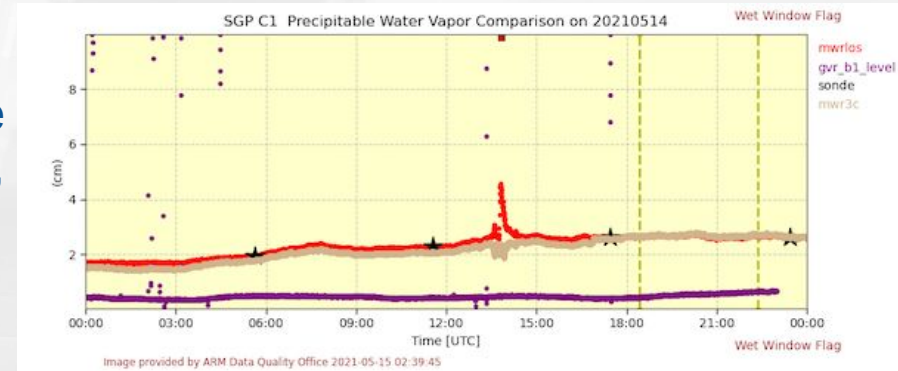
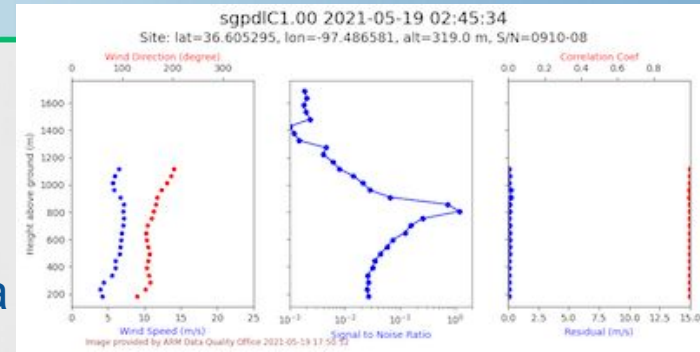
- web-services for downloading data from ARM, Iowa Mesonet ASOS, and Dept. of Agriculture CropScape
- ability to read multiple netCDF4 files with Xarray method
 - ◆ automatically converts time to native format
- ability to read multiple csv and other ASCII data using Pandas
 - ◆ automatically converts time to native format, fills in NaN values, guesses data type
- wrappers around GML csv files to implement README metadata into attributes

So what, you can read a csv file. Big whoop. Why does that matter?

- Converting time and adding metadata to variable attributes means the data is actually correct with units, time zones, and other critical information.

Retrievals and Corrections

- **wind speed profile** from doppler lidar using VAD
- calculating **cloud base height** from lidar/radar data using a basic sobel filter and thresholding
- calculate **sea surface skin temperature** from infrared sky and surface temperature
- convert longwave **radiance to temperature**
- calculate **precipitable water vapor** from radiosonde
- **correct micro-pulse lidar** for data before laser firing, background signal, afterpulse, range and overlap corrections
- correct wind speed and direction for **ship motion**



Quality Control Variables and Filtering

The ARM logo consists of the letters "ARM" in a bold, blue, sans-serif font. Below the letters is a thin, blue, curved line that arches under the text.

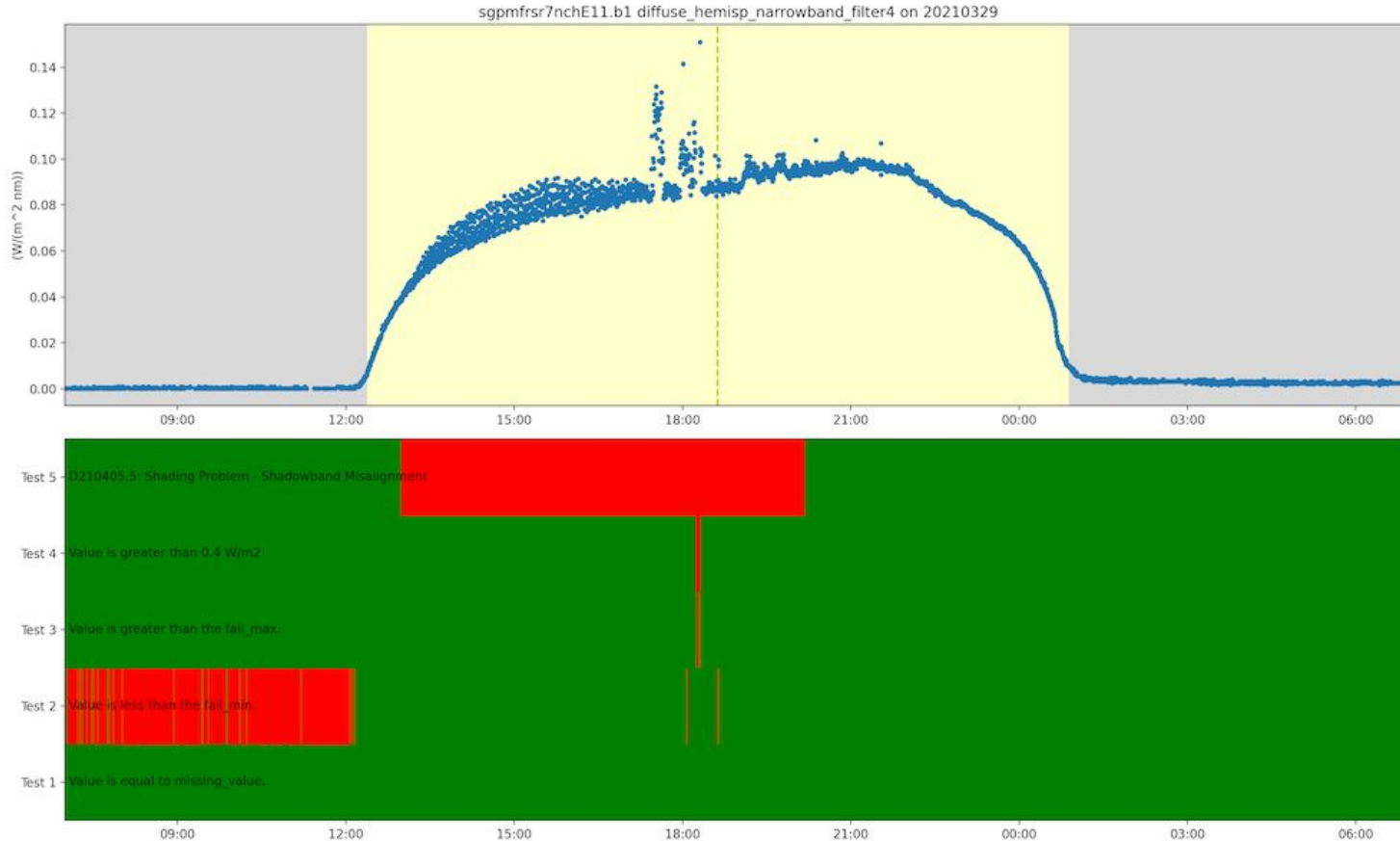
Everyone stores quality control information differently (and have strong feelings about it)

ACT implemented advanced C&F standards for ancillary quality control variables

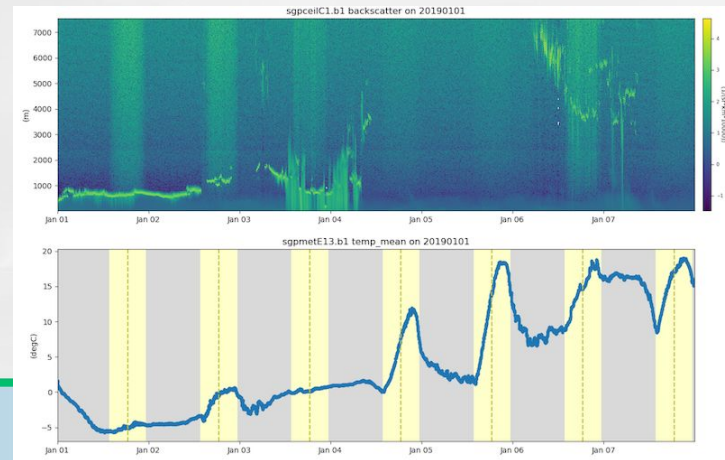
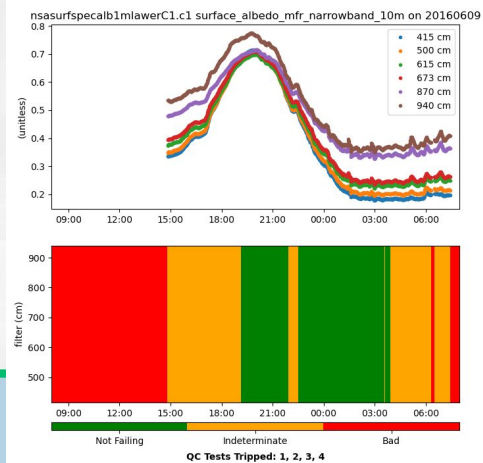
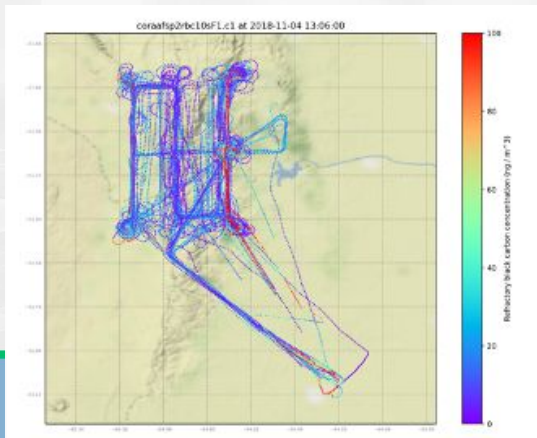
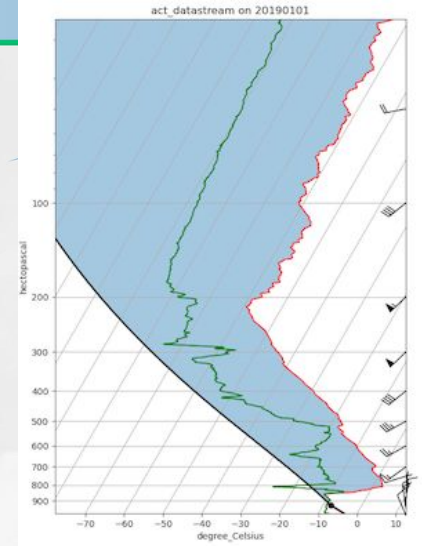
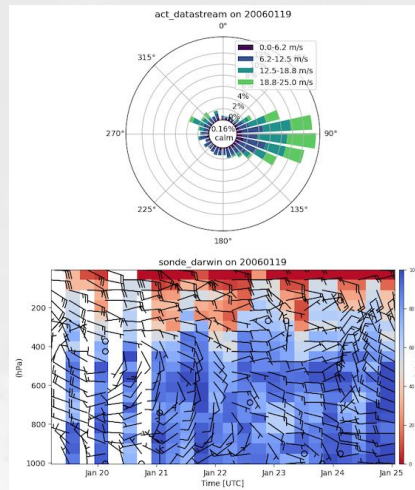
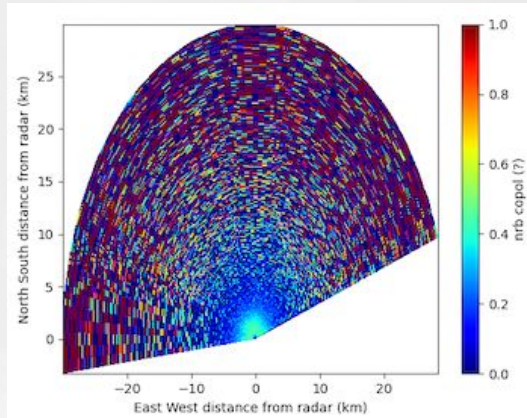
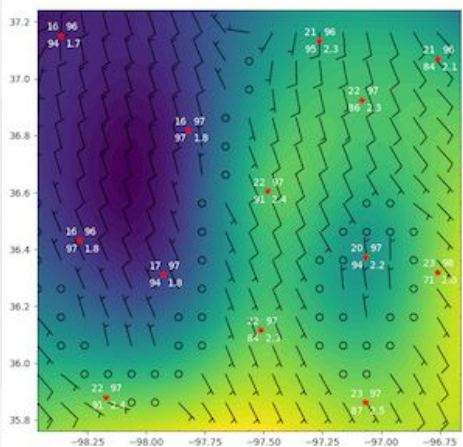
- function to create and link ancillary quality control variable
- function to update other data file styles to C&F standard
- uses bit packing to save space and allow simultaneous flags
- implements assessment categorizing for grouping quality level
- 14 tests currently implemented, more on the way
- filters data based on individual test result or assessment category

The basic takeaway is that you don't need to understand how the quality control is stored, **just set it and forget it.**

Visualizing quality control information



Plotting, where ACT excels



Using Xarray extensions



Search the docs ...

FOR USERS

Getting Started

User Guide

Gallery

Toy weather data

Calculating Seasonal Averages from Time Series of Monthly Means

Compare weighted and unweighted mean temperature

Working with Multidimensional Coordinates

Visualization Gallery

ROMS Ocean Model Example

GRIB Data Example

Applying unvectorized functions with apply_ufunc

Tutorials & Videos

API Reference

How do I ...

Ecosystem

FOR DEVELOPERS/CONTRIBUTORS

Contributing Guide

Xarray Internals

Development Roadmap

You can run this notebook in a [live session](#) [launch binder](#) or view it on [Github](#).

Calculating Seasonal Averages from Time Series of Monthly Means

Author: [Joe Hamman](#)

The data used for this example can be found in the [xarray-data](#) repository. You may need to change the path to [rasm.nc](#) below.

Suppose we have a netCDF or [xarray.Dataset](#) of monthly mean data and we want to calculate the seasonal average. To do this properly, we need to calculate the weighted average considering that each month has a different number of days.

```
[1]: %matplotlib inline
import numpy as np
import pandas as pd
import xarray as xr
import matplotlib.pyplot as plt
```

Open the Dataset

```
[2]: ds = xr.tutorial.open_dataset('rasm').load()
ds
```

[2]: xarray.Dataset

► Dimensions: (time: 36, x: 275, y: 205)

▼ Coordinates:

| | | | | |
|------|--------|---------|-------------------------------------|--|
| time | (time) | object | 1980-09-16 12:00:00 ... 1983-08-... | |
| xc | (y, x) | float64 | 189.2 189.4 189.6 ... 17.15 16.91 | |
| yc | (y, x) | float64 | 16.53 16.78 17.02 ... 27.76 27.51 | |

▼ Data variables:

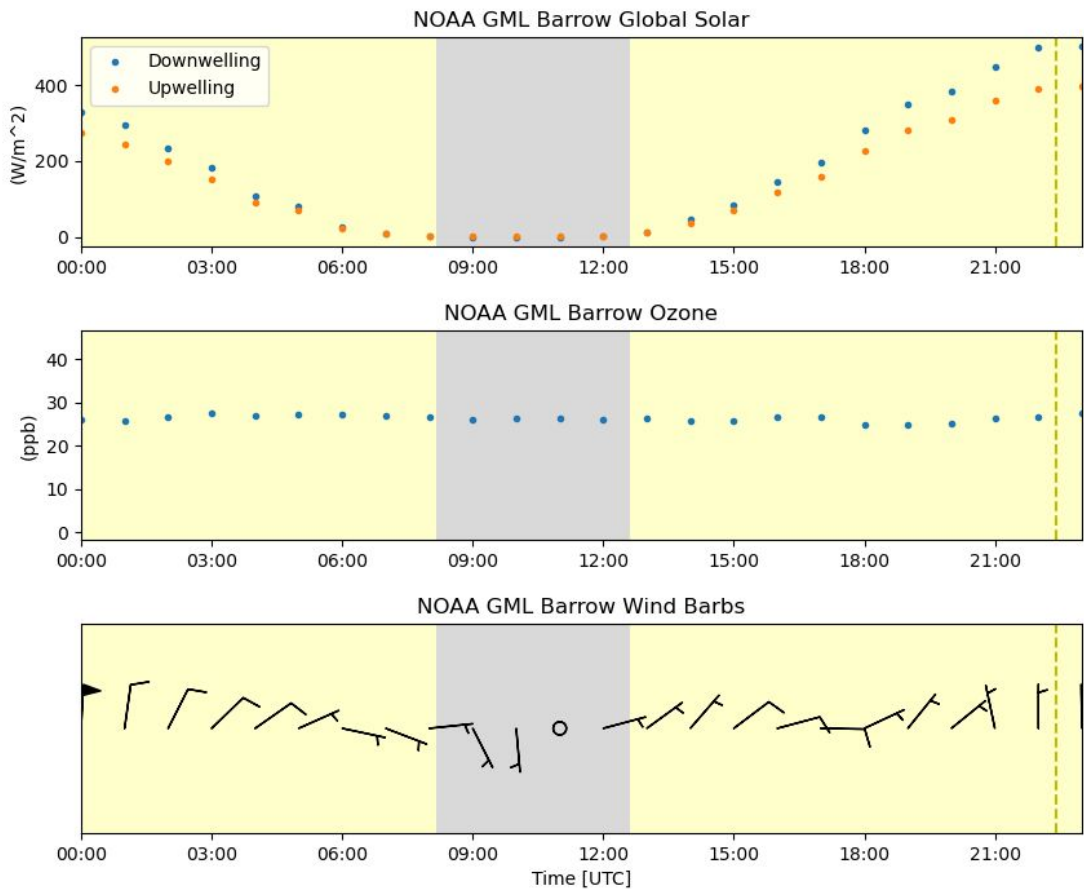
| | | | | |
|------|--------------|---------|-----------------------------------|--|
| Tair | (time, y, x) | float64 | nan nan nan ... 28.66 28.19 28.21 | |
|------|--------------|---------|-----------------------------------|--|

► Attributes: (11)

Xarray is very powerful and utilizes Pandas. Soon you'll be saying **“Wait, Xarray can do that!? Wow, that's amazing.”**

GML Specific Examples

ARM



Example with Barrow Data.

Only 28 lines of code from reading data to displaying plot in a pop-up window.

Is a function missing? Create an issue on github to request, or better yet add it to the library yourself.

<https://github.com/ARM-DOE/ACT/issues>