Automate your climate & weather data analysis with **aospy**

Spencer Hill | Spencer Clark

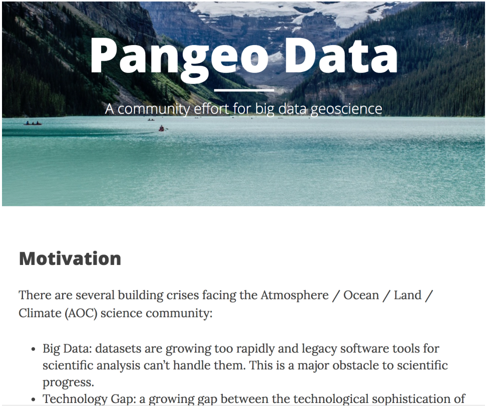
UCLA AOS/Caltech GPS | Princeton AOS

But first | `pangeo-data`: towards scalable climate data analysis tools

`pangeo-data.github.io`
Organized by Ryan Abernathey

Building next-gen, scalable climate research software
All Python, xarray and dask

Looking for more contributors!



Motivation

There are several building crises facing the Atmosphere / Ocean / Land / Climate (AOC) science community:

- Big Data: datasets are growing too rapidly and legacy software tools for scientific analysis can't handle them. This is a major obstacle to scientific progress.
- Technology Gap: a growing gap between the technological sophistication of

Screenshot of pangeo-data website

Motivation | Modern weather/climate research requires lots of data

Legend: *general aospy description in black*
My PhD thesis work as example in gray

Multiple models, simulations, variables, date ranges, etc. of interest
17 climate models, >100 simulations, ∼20 key variables, etc.

Can't perform all desired calculations without automation
1000s of unique calculations desired

Motivation | But automating analyses stymied by seemingly trivial details

E.g. different variable names and grids across models

`lat_bounds` v. `latb`, 10 hPa v. 1 hPa model top

And even if you do, can't keep track of resulting deluge of output

Pre-aospy: directories full of e.g. `precip01.dat`: no metadata!

Safer to just re-do the calculations

Automate your climate & weather data analysis with **aospy**

What it is and how it works

Where’s your data?
What all do you want to compute?
Here are the results.

aospy’s future: join us!

As a user or developer

How it works | Separate description of (1) data on disk vs. (2) individual calculation parameters

Specifics of your data: Use built-in `Proj`, `Model`, `Run` classes
Only need to specify 1 time, 1 place

Physical quantities and regions of interest:
built-in `Var` and `Region` classes
Also need to specify only once

Precise specifications to perform a particular calculation:
built-in `Calc` objects:
Need each time a calculation is performed

Calculate | Included "main" script permutes over all user-specified parameter settings

User: specify all the parameters variations you want

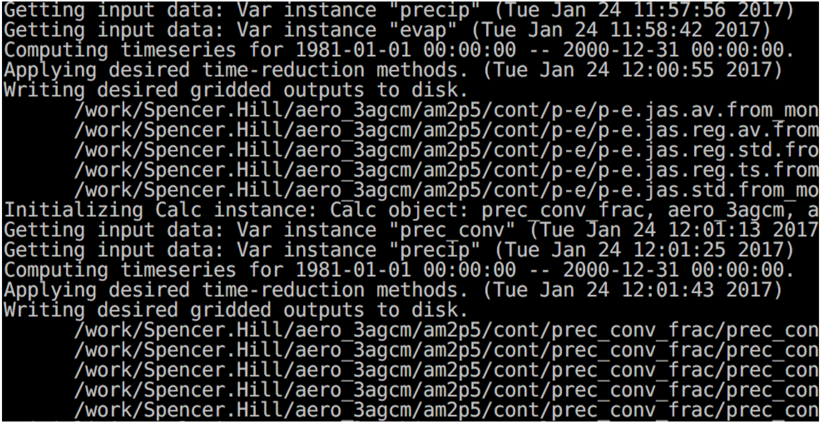
E.g. moist static energy and moist static stability in the control simulation of all my models, averaged over each gridpoint and over these 20 regions, computing monthly and JJA averages and standard deviations, over the default time period for each model, averaged over the column and outputted at each level, using input data on the model-native vertical coordinates and interpolated to pressure levels

aospy permutes over all of them,
generating a `Calc` for each and executing it

In parallel!

Find the results

Calculation results stored in simple, metadata-rich directory structure

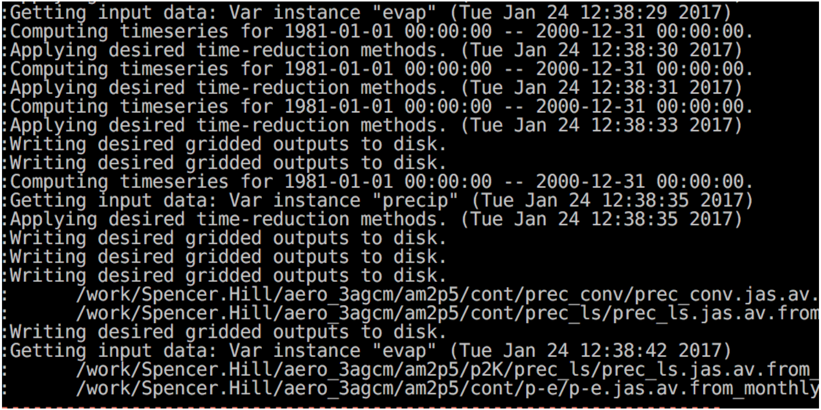


```
Getting input data: Var instance "precip" (Tue Jan 24 11:57:56 2017)
Getting input data: Var instance "evap" (Tue Jan 24 11:58:42 2017)
Computing timeseries for 1981-01-01 00:00:00 -- 2000-12-31 00:00:00.
Applying desired time-reduction methods. (Tue Jan 24 12:00:55 2017)
Writing desired gridded outputs to disk.
      /work/Spencer.Hill/aero_3agcm/am2p5/cont/p-e/p-e.jas.av.from_mon
      /work/Spencer.Hill/aero_3agcm/am2p5/cont/p-e/p-e.jas.reg.av.from
      /work/Spencer.Hill/aero_3agcm/am2p5/cont/p-e/p-e.jas.reg.std.fro
      /work/Spencer.Hill/aero_3agcm/am2p5/cont/p-e/p-e.jas.reg.ts.from
      /work/Spencer.Hill/aero_3agcm/am2p5/cont/p-e/p-e.jas.std.from_mo
Initializing Calc instance: Calc object: prec_conv_frac, aero_3agcm, a
Getting input data: Var instance "prec_conv" (Tue Jan 24 12:01:13 2017
Getting input data: Var instance "precip" (Tue Jan 24 12:01:25 2017)
Computing timeseries for 1981-01-01 00:00:00 -- 2000-12-31 00:00:00.
Applying desired time-reduction methods. (Tue Jan 24 12:01:43 2017)
Writing desired gridded outputs to disk.
      /work/Spencer.Hill/aero_3agcm/am2p5/cont/prec_conv_frac/prec_con
      /work/Spencer.Hill/aero_3agcm/am2p5/cont/prec_conv_frac/prec_con
      /work/Spencer.Hill/aero_3agcm/am2p5/cont/prec_conv_frac/prec_con
      /work/Spencer.Hill/aero_3agcm/am2p5/cont/prec_conv_frac/prec_con
      /work/Spencer.Hill/aero_3agcm/am2p5/cont/prec_conv_frac/prec_con
```

Figure: Screenshot of logging information printed by aospy during main script execution

Find output data

Calculation results stored in simple, metadata-rich directory structure



```
:Getting input data: Var instance "evap" (Tue Jan 24 12:38:29 2017)
:Computing timeseries for 1981-01-01 00:00:00 -- 2000-12-31 00:00:00.
:Applying desired time-reduction methods. (Tue Jan 24 12:38:30 2017)
:Computing timeseries for 1981-01-01 00:00:00 -- 2000-12-31 00:00:00.
:Applying desired time-reduction methods. (Tue Jan 24 12:38:31 2017)
:Computing timeseries for 1981-01-01 00:00:00 -- 2000-12-31 00:00:00.
:Applying desired time-reduction methods. (Tue Jan 24 12:38:33 2017)
:Writing desired gridded outputs to disk.
:Writing desired gridded outputs to disk.
:Computing timeseries for 1981-01-01 00:00:00 -- 2000-12-31 00:00:00.
:Getting input data: Var instance "precip" (Tue Jan 24 12:38:35 2017)
:Applying desired time-reduction methods. (Tue Jan 24 12:38:35 2017)
:Writing desired gridded outputs to disk.
:Writing desired gridded outputs to disk.
:Writing desired gridded outputs to disk.
:      /work/Spencer.Hill/aero_3agcm/am2p5/cont/prec_conv/prec_conv.jas.av.
:      /work/Spencer.Hill/aero_3agcm/am2p5/cont/prec_ls/prec_ls.jas.av.from
:Writing desired gridded outputs to disk.
:Getting input data: Var instance "evap" (Tue Jan 24 12:38:42 2017)
:      /work/Spencer.Hill/aero_3agcm/am2p5/p2K/prec_ls/prec_ls.jas.av.from_
:      /work/Spencer.Hill/aero_3agcm/am2p5/cont/p-e/p-e.jas.av.from_monthly
```

Figure: Same, but jobs submitted in parallel. Massive speed-up but logging needs work!

Tech specs | Adhering to modern best practices in open-source software

Fully open-source w/ code hosted on Github

`github.com/spencerahill/aospy`

Support for wide range of platforms: Python 2.7, 3.4, 3.5, 3.6; Linux, MacOS/OS X, and Windows

Works on laptops/desktops and large clusters

v0.1 released last night: `pip install aospy`

Coming soon: (`conda -c conda-forge install aospy`

Tech specs | Adhering to modern best practices in open-source software

Testing & continuous integration: maintain code quality

Travis CI for Linux/Mac; AppVeyor for Windows; Coveralls for test coverage

Documentation: hosted on ReadTheDocs

`aospy.readthedocs.io`

Open science

aospy promotes reproducible research and improves code quality

Enables writing clearer code

`xarray` hugely important for this

Isolate code describing your particular data from code describing the quantities you are trying to compute

Use both to more easily share your work with colleagues and journals

Future | Eager for more users and new contributors

Current user base: me and Spencer Clark

Please be our third!

Both use `aospy` for all of our research

Has fueled research insights otherwise unattainable

Current developer base: also me and Spencer Clark

Please be our third!

Best place to start: `aospy.readthedocs.io`