

Demonstrating cloud-based remote sensing data workflows with xarray

Emma Marshall
University of Utah, National Center for Atmospheric Research

MOTIVATION

Recent advances in satellite imagery availability, cloud-computing resources and open-source software necessitate the need for detailed, accessible educational resources in order to fully realize the public benefit and scientific potential of these resources.

Mentors: Deepak Cherian (NCAR), Scott Henderson (University of Washington), Jessica Scheick (University of New Hampshire), Kevin Paul (NCAR)

BACKGROUND

- Cloud computing resources can democratize scientific participation, reduce computational barriers to entry
- Accessing, manipulating data is a common bottleneck in remote sensing research workflows
- Xarray has functionality for organizing, analyzing raster data, and backend integration with cloud-optimized data types

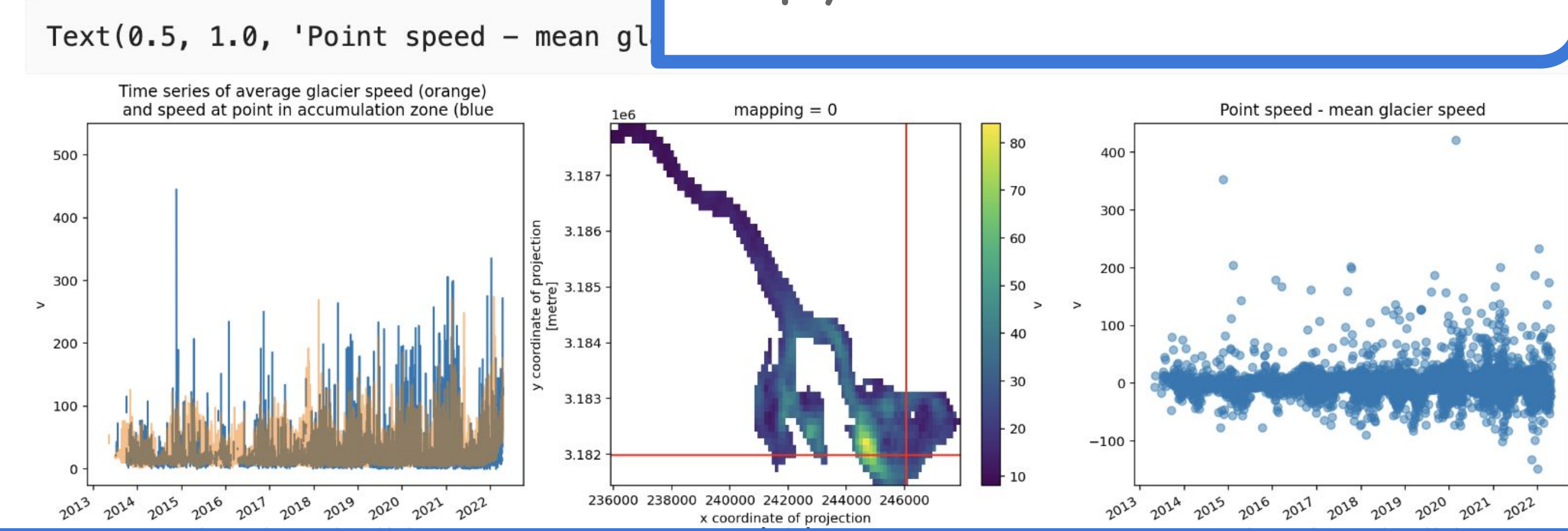
JUPYTER BOOK TUTORIALS

- Show data access steps for multiple cloud-hosting platforms
- Use accessible, explanatory text
- Include errors and solutions (not just showing what works!)
- Focus on manipulating data to form data cubes
- Demonstrate parallelized workflows
- Emphasizing and supporting open, reproducible science

Extracting and visualizing data at a single point

We can use xarray's `.sel()` to extract velocity data at a single point or within a subset along given dimensions. In this example, we use `.sel()` to compare the magnitude of velocity of ice flow at a point in the glacier's accumulation zone to the mean magnitude of velocity of the entire glacier.

```
fig, axes = plt.subplots(ncols=3, figsize=(20,5))
sample_glacier_raster.v.sel(x = 246052.5, y = 3181987.5).plot(ax=axes[0]);
sample_glacier_raster.v.mean(dim=['x','y']).plot(ax=axes[1], alpha = 0.5);
sample_glacier_raster.v.mean(dim='mid_date').plot(ax=axes[2]);
axes[1].axvline(x=246052.5, c='red')
axes[1].axhline(y=3181987.5, c='red')
(sample_glacier_raster.v.sel(x = 246052.5, y = 3181987.5)).plot(ax=axes[0], alpha = 0.5);
axes[0].set_title('Time series of average speed at point in accumulation zone (blue)')
axes[2].set_title('Point speed - mean glacier speed')
```



Check out the Jupyter book!

Book 1: Accessing and analyzing ITS_LIVE ice velocity dataset stored in S3 buckets on Amazon Web Services

OBJECTIVES:

- Gain skills and experience working with cloud computing resources and parallelizing workflows
- Develop educational resources and make open-source contributions to support the processing of cloud-hosted remote sensing data with xarray

Book 2: Time series analysis of backscatter variability over proglacial lakes using synthetic aperture radar imagery hosted by Microsoft Planetary Computer

OPEN SOURCE CONTRIBUTIONS & INVOLVEMENT

Netcdf data cleaning example for xarray-tutorial, documentation contributions to xarray-contrib, dataset contribution to xarray-pydata, co-presented @ SciPy 2022 xarray tutorial, Austin, TX July 2022

Re-organize InSAR ice velocity data

This is an example of cleaning data accessed in netcdf format and preparing it for a Jupyter notebook.

The dataset we will use contains InSAR-derived ice velocity for the Embayment in Antarctica. The data is downloaded from: <https://github.com/SciTools/xarray-pydata> but this example uses only a subset of the full dataset.

Downloaded data is `.hdr` and `.dat` files for each year, and a `.nc` file for each year.

The `.nc` object is a dataset with dimensions `x,y` and data variables `vx,vy,err` vars. We'd like to re-organize this so that there are `vx,vy` along a time dimension.

```
import xarray as xr
import pandas as pd
import os
import numpy as np
```

WHAT DID I LEARN?

- Collaborative workflows
- Vectorized and xarray-native programming
- Working with cloud-hosted data and cloud computing resources
- Better data sharing practices
 - Jupyter book, github gists, notebook{sharing}.space

