Scalable and Computationally Reproducible Approaches to Arctic Research

Matt Jones, Bryce Mecum, Jeanette Clark, Sam Csik

September 19, 2022

Table of contents

Preface

About

This 5-day in-person workshop will provide researchers with an introduction to advanced topics in computationally reproducible research in python and R, including software and techniques for working with very large datasets. This includes working in cloud computing environments, docker containers, and parallel processing using tools like parsl and dask. The workshop will also cover concrete methods for documenting and uploading data to the Arctic Data Center, advanced approaches to tracking data provenance, responsible research and data management practices including data sovereignty and the CARE principles, and ethical concerns with data-intensive modeling and analysis.

Schedule

Code of Conduct

Please note that by participating in this activity you agree to abide by the NCEAS Code of Conduct.

	Monday	Tuesday	Wednesday	Thursday	Friday
08:00-08:30	Coffee (optional)	Coffee (optional)	Coffee (optional)	Coffee (optional)	Coffee (optional)
08:30-09:00	**1. Welcome and Course Overview** (Jeanette)	**6. Data structures and formats for large data** (Bryce)	**10. Spatial and Image Data using GeoPandas** (Jeanette)	**15. Google Earth Engine** (Ingmar, Sam)	**19. What is cloud computing anyways**? (Matt)
09:00-09:30					
09:30-10:00	**2. Remote computing** (Sam)		**11. Data futures: Parquet and Arrow** (Jeanette)		
10:00-10:30					
10:30-11:00	BREAK	BREAK	BREAK	BREAK	BREAK
11:00-11:30	**3. Python programming on clusters** (Jeanette)	**7. Parallelization with Dask** (Bryce)	**12. Software Design II** (Bryce)	**16. Billions of Ice Wedge Polygons** (Chandi)	**20. Reproducibility redux via containers** (Bryce)
11:30-12:00					**Survey Feedback Q & A**
12:00-12:30	Lunch	Lunch	Lunch	Lunch	Adjourn
12:30-13:00					
13:00-13:30	**4. Pleasingly Parallel Programming** (Matt)	**8. Group project I Data staging and pre-processing** (Jeanette)	**13. Group project II Parallel data processing** (Jeanette)	**17. Group project III Visualizing big geospatial data** (Jeanette)	
13:30-14:00					
14:00-14:30					
14:30-15:00					
15:00-15:30	Break	Break	Break	Break	
15:30-16:00	**5. Documenting and Publishing Data** (Daphne)	**9. Software design I** (Bryce)	**14. Data Ethics** (Matt)	**18. Workflows for data staging and publishing** (Jeanette)	
16:00-16:30			**Breather** Catch-up		
16:30-17:00	Q&A	Q&A	Q&A	Q&A	

Setting Up

In this course, we will be using Python (> 3.0) as our primary language, and VS Code as our IDE. Below are instructions on how to get VS Code set up to work for the course. If you are already a regular Python user, you may already have another IDE set up. We strongly encourage you to set up VS Code with us, because we will use your local VS Code instance to write and execute code on one of the NCEAS servers.

Download VS Code and Extensions

First, download VS Code if you do not already have it installed.

Check to make sure you have Python installed if you aren't sure you do. To do this, from the terminal run:

```
python3 --version
```

If you get an error, it means you need to install Python. Here are instructions for getting installed, depending on your operating system. Note: There are many ways to install and manage your Python installations, and advantages and drawbacks to each. If you are unsure about how to proceed, feel free to reach out to the instructor team for guidance.

- Windows: Download and run an installer from Python.org.
- Mac: Install using homebrew. If you don't have homebrew installed, follow the instructions from their webpage.

    - `brew install python3`

After you run your install, make sure you check that the install is on your system PATH by running `python3 --version` again.

Set up VS Code

This section summarizes the official VS Code tutorial. For more detailed instructions and screenshots, see the source material

First, install the Python extension for VS Code.

Open a terminal window in VS Code from the Terminal drop down in the main window. Run the following commands to initialize a project workspace in a directory called `training`. This example will show you how to do this locally. Later, we will show you how to set it up on the remote server with only one additional step.

```
mkdir training
cd training
code .
```

Next, we will select the Python interpreter for the project. Open the **Command Palette** using Command + Shift + P (Control + Shift + P for windows). The Command Palette is a handy tool in VS Code that allows you to quickly find commands to VS Code, like editor commands, file edit and open commands, settings, etc. In the Command Palette, type "Python: Select Interpreter." Push return to select the command, and then select the interpreter you want to use (your Python 3.X installation).

Finally, download the Jupyter extension. You can create a test Jupyter Notebook document from the command pallete by typing "Create: New Jupyter Notebook" and selecting the command. This will open up a code editor pane with a notebook that you can test.

Test your local setup (Optional)

To make sure you can write and execute code in your project, create a Hello World test file.

- From the File Explorer toolbar, or using the terminal, create a file called `hello.py`

6

- Add some test code to the file, and save

```
msg = "Hello World"
print(msg)
```

- Execute the script using either the Play button in the upper-right hand side of your window, or by running `python3 hello.py` in the terminal.
    - For more ways to run code in VS Code, see the [tutorial](#)

About this book

These written materials reflect the continuous development of learning materials at the Arctic Data Center and NCEAS to support individuals to understand, adopt, and apply ethical open science practices. In bringing these materials together we recognize that many individuals have contributed to their development. The primary authors are listed alphabetically in the citation below, with additional contributors recognized for their role in developing previous iterations of these or similar materials.

This work is licensed under a [Creative Commons Attribution 4.0 International License](#).

Citation: Matthew B. Jones, Bryce Mecum, S. Jeanette Clark, Samantha Csik. 2022. Scalable and Computationally Reproducible Approaches to Arctic Research.

Additional contributors: Amber E. Budden, Natasha Haycock-Chavez, Noor Johnson, Stephanie Hampton, Jim Regetz, Bryce Mecum, Julien Brun, Julie Lowndes, Erin McLean, Andrew Barrett, David LeBauer, Jessica Guo.

This is a Quarto book. To learn more about Quarto books visit https://quarto.org/docs/books.

1 Welcome and Introductions

Jeanette Clark

This course is one of three that we are currently offering, covering fundamentals of open data sharing, reproducible research, ethical data use and reuse, and scalable computing for reusing large data sets.

Fundamentals in Data Management for Qualitative and Quantitative Arctic Research
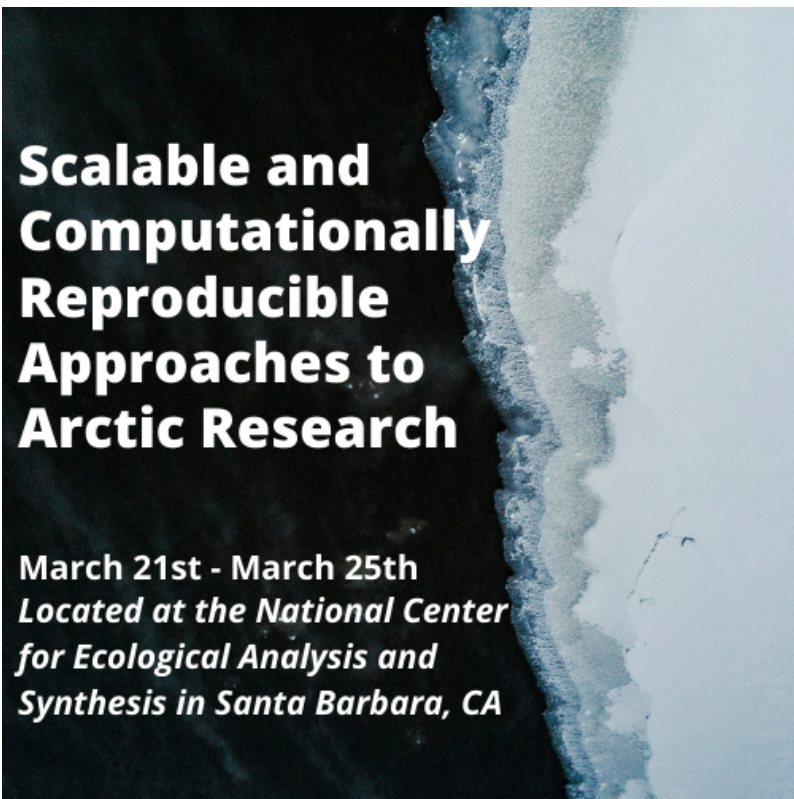
April 18th - April 22nd
Located at the National Center for Ecological Analysis and Synthesis in Santa Barbara, CA

Reproducible Practices for Arctic Research Using R

February 14th - February 18th, 2022
This course will be taught virtually

2 Remote Computing

Sam Csik

2.1 Introduction

3 Python Programming on Clusters

Jeanette Clark

- Basic Python review
- Writing in Jupyter notebooks
- Writing functions in Python

3.1 Introduction

- VS Code + remote development on a cluster is easy and way faster than your local machine
- Jupyter is a great way to do literate analysis
- Functions provide ways to reuse your code across notebooks/projects

3.2 Python on the cluster

- Connect to the server
- Start a `training` project and pick interpreter (this could also go in Sam's session)
- Create and execute `hello.py`

  - from the IDE as a whole
  - from IDE line by line
  - from the terminal

3.3 Jupyter notebooks

- Create a notebook
- Load in some libraries (pandas, numpy, scipy, matplotlib)
- Read in a csv
- group and summarize by a variable
- create a simple plot

3.4 Functions

- create `myplot.py`
- write `myplot()` function to create the same plot we did in section above
- load myplot into jupyter notebook (`from myplot.py import myplot`)
- replace old plot method with new function

3.5 Resources

4 Pleasingly Parallel Programming

Matt Jones

4.1 Introduction

5 Documenting and Publishing Data

Daphne Virlar-Knight, Natasha Haycock-Chavez, Amber Budden, Matt Jones

5.1 Introduction

6 Spatial and Image Data Using GeoPandas

Jeanette Clark"

```python
import geopandas as gpd
import rasterio as rio
import requests
import matplotlib.pyplot as plt
```

```python
url = 'https://cn.dataone.org/cn/v2/resolve/urn:uuid:dd61089d-f50e-4d87-9b75-6b4e2bd24776'

response = requests.get(url)
open("Coastal_2020_12.tif", "wb").write(response.content)
```
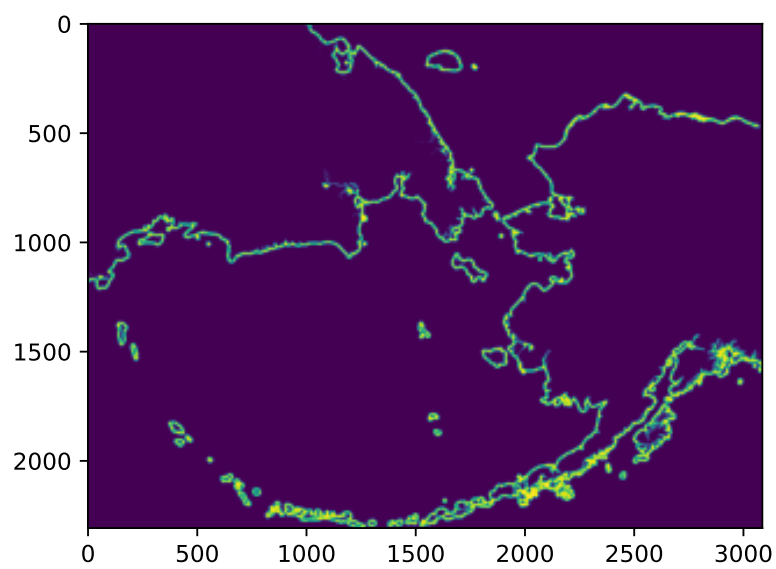
1132748

```python
with rio.open("Coastal_2020_12.tif") as dem_src:
    dtm_pre_arr = dem_src.read(1)

plt.imshow(dtm_pre_arr)
```

<matplotlib.image.AxesImage at 0x11cbeed60>

References