

Open Climate Science for Agriculture: A Reproducible Computational Science Curriculum

K. Arthur Endsley

Numerical Terradynamic Simulation Group (NTSG)

W.A. Franke College of Forestry and Conservation

April 18, 2024



Need for Reproducible Science Training

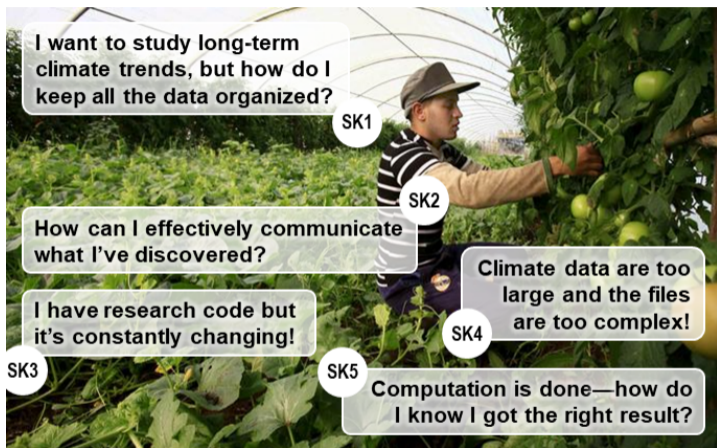
- Vast majority of domain scientists who use computer-aided inference are **self-taught**

Need for Reproducible Science Training

- Vast majority of domain scientists who use computer-aided inference are **self-taught**
- CMIP6 produced an estimated **20-40 petabytes** of data
- Climate scientists are spending more of their time on **downloading data, debugging code, and solving compatibility issues**

Need for Reproducible Science Training (2)

- Vast majority of domain scientists who use computer-aided inference are **self-taught**
- CMIP6 produced an estimated **20-40 petabytes** of data
- Climate scientists are spending more of their time on **downloading data, debugging code, and solving compatibility issues**



Transform to Open Science (TOPS)

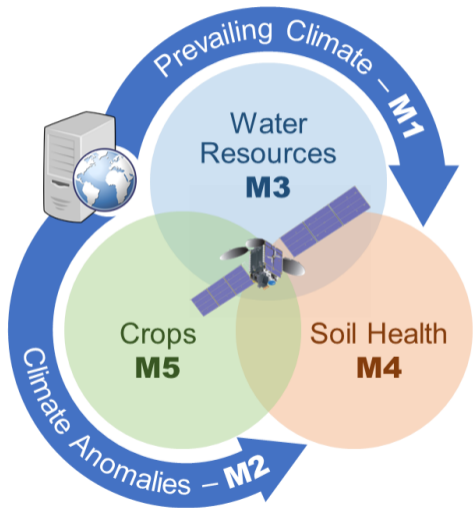
In 2023, The White House declares the "Year of Open Science" and **NASA launches the online Open Science 101 curriculum**:

- Ethos of Open Science
- Open Tools and Resources
- Open Data
- Open Code
- Open Results

<http://nasa.github.io/Transform-to-Open-Science>



Our NASA-Funded Curriculum



“Open Climate Science for Agriculture”

- **M1:** "Open Climate Data" (*How and where to access NASA climate data*)
- **M2:** "Computational Climate Science" (*How to calculate climate indices*)
- **M3:** "Water Resources" (and Water Deficits)
- **M4:** "Soil Health"
- **M5:** "Crop Conditions and Crop Production"

2-year NASA-funded project; to be translated into French and Arabic

Learning Outcomes

Revising our proposal, we now envision four *Core Competencies* (CC) for computational science:

CC1: Project & Data Management

- e.g., Navigating a file system; Appropriate naming of files and directories; File compression; Backups; Metadata

Learning Outcomes (2)

Revising our proposal, we now envision four *Core Competencies* (CC) for computational science:

CC1: Project & Data Management

CC2: Scientific Programming

- e.g., Understanding machine numeric data types and arrays; Resource profiling; Concurrency; Debugging

Learning Outcomes (3)

Revising our proposal, we now envision four *Core Competencies* (CC) for computational science:

CC1: Project & Data Management

CC2: Scientific Programming

CC3: Collaborative Computational Science

- e.g., Source control management (“Version control”); Publishing code and data; Legible coding styles; Reading API documentation; Literate programming

Learning Outcomes (4)

Revising our proposal, we now envision four *Core Competencies* (CC) for computational science:

CC1: Project & Data Management

CC2: Scientific Programming

CC3: Collaborative Computational Science

CC4: Sustainable Computational Science

- e.g., Reproducible workflows; Software releases and semantic versioning; Runtime input parameters; Verification; Containerization

<https://github.com/OpenClimateScience/Core-Competencies>

Impact Assessment

CC3	3.1 Source Control Management	BOTH: "I regularly use version control to track changes to notes, documents, or research code."
		BOTH: "If I wanted to undo some changes made to my research code, I can easily recover a previous version."
		BOTH: "If there's a bug in my code, I can figure out when it was introduced and what versions of my software it affects."
		RPT: "I know how to use SCM software like Git, Mercurial, or SVN to track changes in code."
		RPT: "I feel confident using distributed SCM software like Git or Mercurial to collaborate with others on the same software repository."
	3.2 Public Repositories	"I regularly publish my code or data in public repositories on the internet."
	"I've used a public repository like SourceForge, Bitbucket, or Github either to report bugs, download software, or network with	

Practicing Reproducible Computational Science

Modules 3-5 will each offer an opportunity to practice packaging and delivering a software project.

M3 (Water Resources)

*Virtual environments
Initializing a Git repo
Relating outputs-inputs
File-level metadata

M4 (Healthy Soils)

*Python packaging, installs
Git branching and merging
Git tags, Github releases
Zenodo integration, DOIs

M5 (Crops)

*Containers
Model verification

How You Can Help!

- **Contribute to the Core Competencies curriculum guide**
- **Suggest regional climate-data case studies**
- **Identify relevant, interesting NASA datasets**
- **Contribute Jupyter Notebooks directly to Module 2**



Follow our Github community:
<https://OpenClimateScience.github.io/>