What we talk about when we talk about scientific programming

PROPL 2025

Patrick Ferris

Monday 13th October, 2025

University of Cambridge

Programming *Scientifically*?

Scientific Method

By focusing on the scientific method, we can extract properties of scientific programming that are more fundamental than exhaustively defining what constitutes a scientific program and what does not [4, 6]!

Falsifiable
Hypothesis

Repeatable
Experimentation

Reproducible
Methodologies

Data, Code, Programmers

The data: Tropical Moist Forests *Redux*

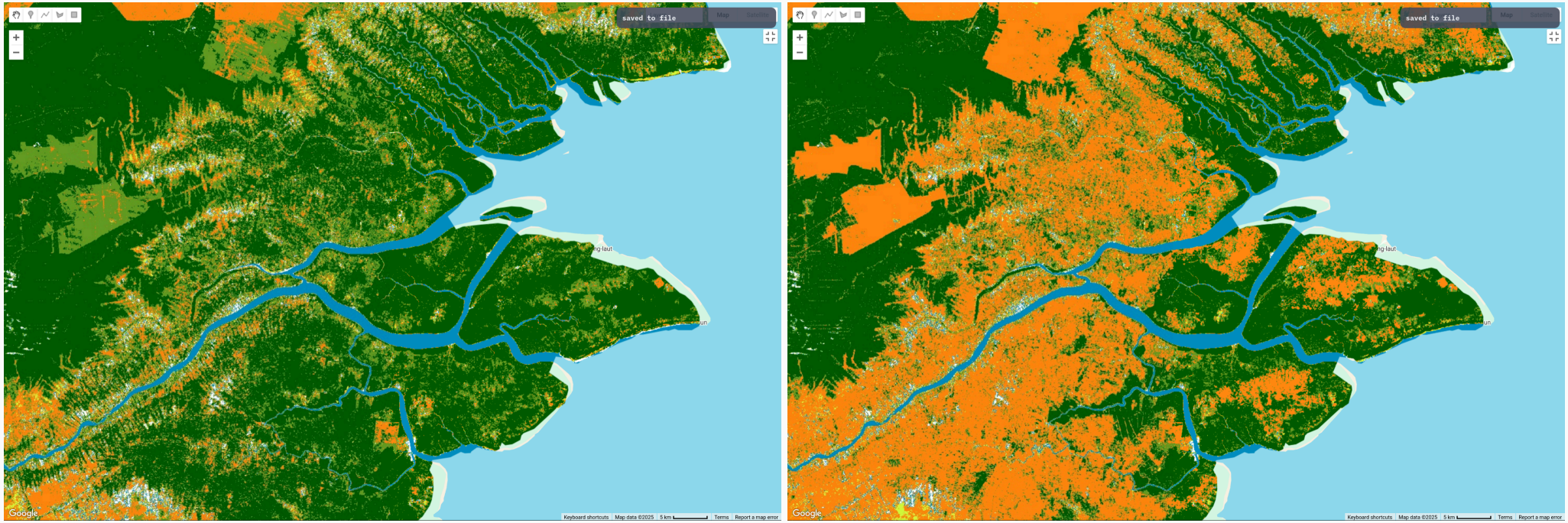
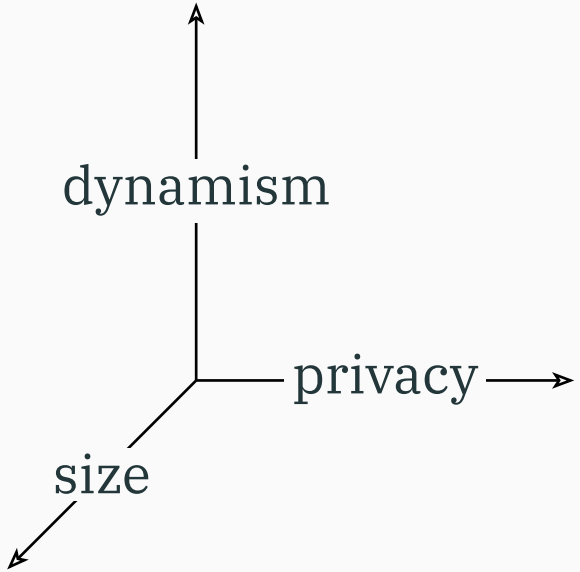


Figure 1: European Commission JRC Tropical Moist Forest Dataset from 2021 (left) and **2024** (right) for the year 2008 in Indonesia [7]. Undisturbed (■), Degraded (■), Deforested (■), Regrowth (■), Water (■) and Other ( )

The data

Our work last year using the Tropical Moist Forest dataset illustrates a need for some form of incremental re-evaluation. Upon re-evaluation the amount of change in land classified as deforestation is now **8.5%** (compared to 3.05% between 2021-2022 datasets)! [2]

If the data changes, what about published conclusions?



The code (1/2)

> Computational work must reflect the committed attitude of experimentalists towards caring about precise, professional, repeatable, meticulous work – no-one with the same casual attitude to experimental instrumentation as many researchers have to code would be allowed anywhere near a lab.
>
> — Baxter et al. [1]

The code (2/2)

In what ways does writing code for scientific programming differ from conventional methods?

> *Literate programming, discretely executable for exploratory workflows.*

Are the ways in which we publish and distribute software amenable to the kinds of falsifiable, repeatable and reproducible experiments that the scientific method requires?

The code (2/2)

In what ways does writing code for scientific programming differ from conventional methods?

> *Literate programming, discretely executable for exploratory workflows.*

Are the ways in which we publish and distribute software amenable to the kinds of falsifiable, repeatable and reproducible experiments that the scientific method requires?

> **NO!**

Vernacular programmers [5]

It is important to recognise that a vast majority of people programming scientifically are not “computer scientists”.

> Although they do not generally have software engineering training, scientists create large software systems that model physical systems to predict future conditions; analyze satellite data; control mobile remote sensing systems; and visualize data to communicate their results with others
>
> — Pertseva et al. “A theory of scientific programming efficacy”[3]

Vernacular programmers [5]

It is important to recognise that a vast majority of people programming scientifically are not “computer scientists”.

> Although they do not generally have software engineering training, scientists create large software systems that model physical systems to predict future conditions; analyze satellite data; control mobile remote sensing systems; and visualize data to communicate their results with others
>
> — Pertseva et al. “A theory of scientific programming efficacy”[3]

We should be building with empathy for whoever our *vernacular programmer* is!

In practice?

```
shelter> echo hello > hello.txt
shelter[main#bbd691a] : { mode: rw }> @ session exp1
shelter[exp1#bbd691a] : { mode: rw }> echo PROPL >> hello.txt
shelter[exp1#dd5bab8] : { mode: rw }> @ session main
shelter[main#bbd691a] : { mode: rw }> echo "to" >> hello.txt
shelter[main#b9abef4] : { mode: rw }> cat hello.txt
hello
to
shelter[main#d1c1728] : { mode: rw }> @ session exp1
shelter[exp1#dd5bab8] : { mode: rw }> @ replay main
shelter[exp1#df1b4e1] : { mode: rw }> cat hello.txt
hello
to
PROPL
```

Listing 1: An example `shelter` session

Bibliography

[1] Robert Baxter, Neil Chue Hong, Dirk Gorissen, James Hetherington, and Ilian Todorov. 2012. The Research Software Engineer. (September 2012).

[2] Patrick Ferris, Michael Dales, Tom Swinfield, Sadiq Jaffer, Srinivasan Keshav, and Anil Madhavapeddy. 2024. Uncertainty at scale: how CS hinders climate research. *Undone Computer Science* (2024).

[3] Elizaveta Pertseva, Melinda Chang, Ulia Zaman, and Michael Coblenz. 2024. A theory of scientific programming efficacy. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, 2024. 1–12.

[4] Karl Popper. 2005. *The logic of scientific discovery*. Routledge.

[5] Mary Shaw. 2022. Myths and mythconceptions: What does it mean to be a programming language, anyhow?. *Proceedings of the ACM on Programming Languages* 4, HOPL (2022), 1–44.

[6] Tim Storer. 2017. Bridging the chasm: A survey of software engineering practice in scientific programming. *ACM Computing Surveys (CSUR)* 50, 4 (2017), 1–32.

[7] C. Vancutsem, F. Achard, J.-F. Pekel, G. Vieilledent, S. Carboni, D. Simonetti, J. Gallego, L. E. O. C. Aragão, and R. Nasi. 2021. Long-term (1990–2019) monitoring of forest cover changes in the humid tropics. *Science Advances* 7, 10 (2021), eabe1603. https://doi.org/10.1126/sciadv.abe1603