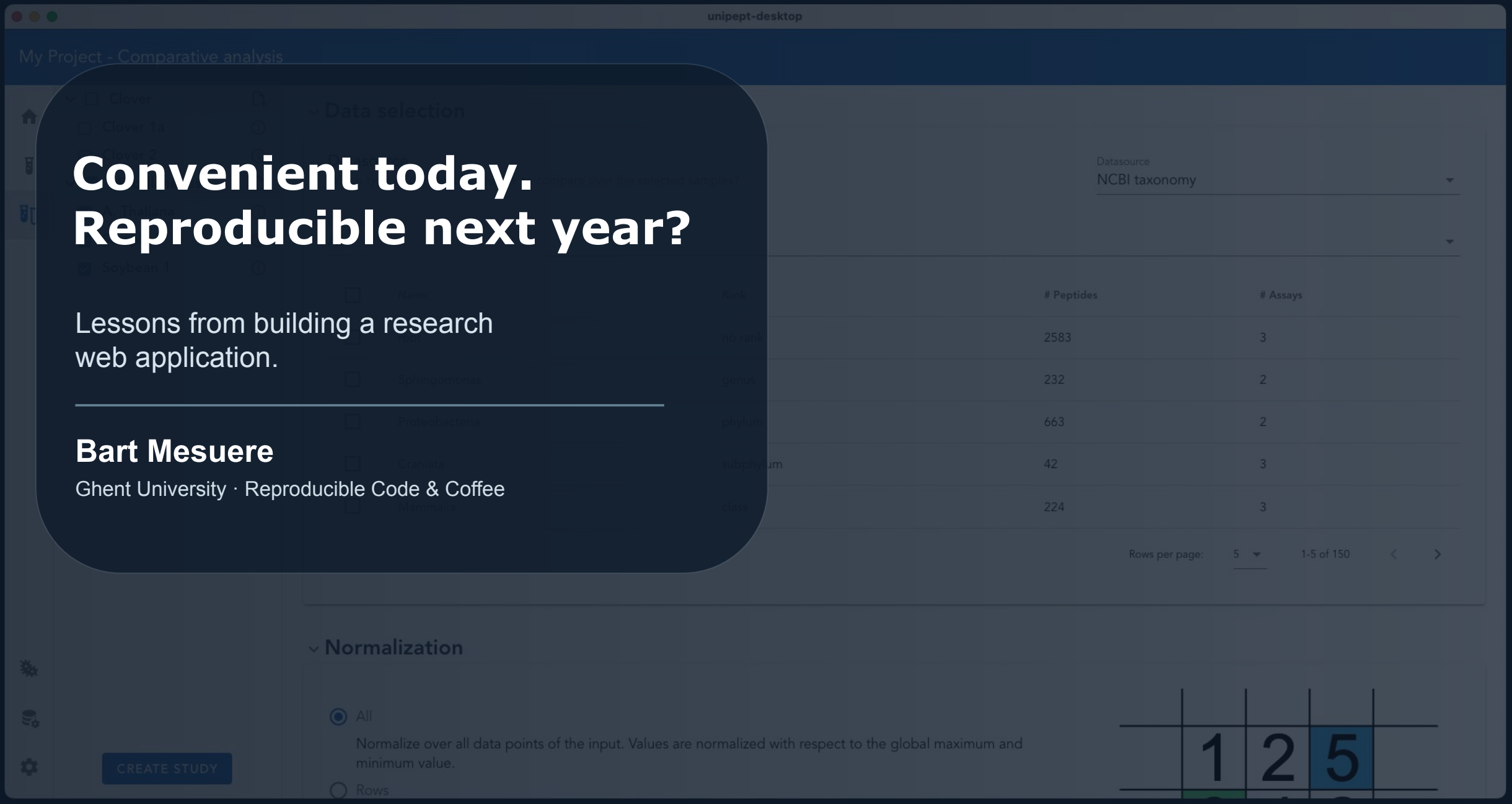
Convenient today. Reproducible next year?

Lessons from building a research web application.

Bart Mesuere
Ghent University · Reproducible Code & Coffee

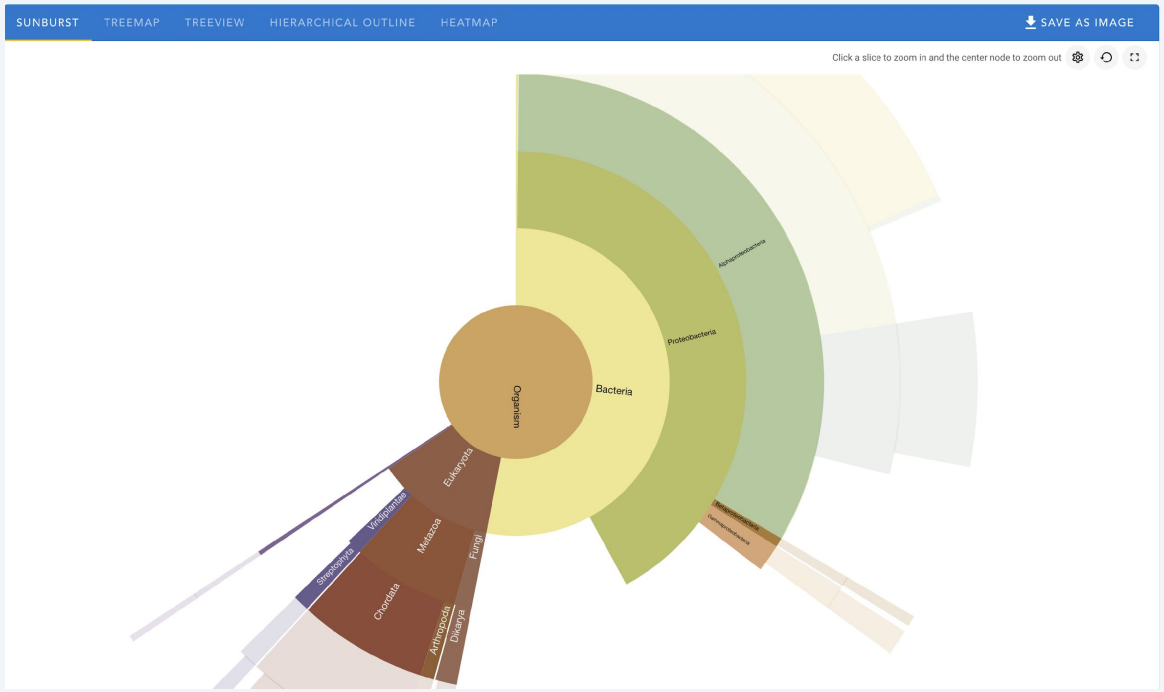


First, what is Unipept?

A web application for analysing protein mixtures using UniProt reference data. The web is a sensible choice here.

Why a web app works well

- **No installation barrier**
  Researchers can start from a browser and a URL.
- **One shared version**
  Bug fixes and improvements are available to everyone immediately.
- **Lower support burden**
  The team maintains one main deployment instead of many local setups.



For day-to-day research use, this is exactly what you want: easy access, one familiar interface, and no installation friction.

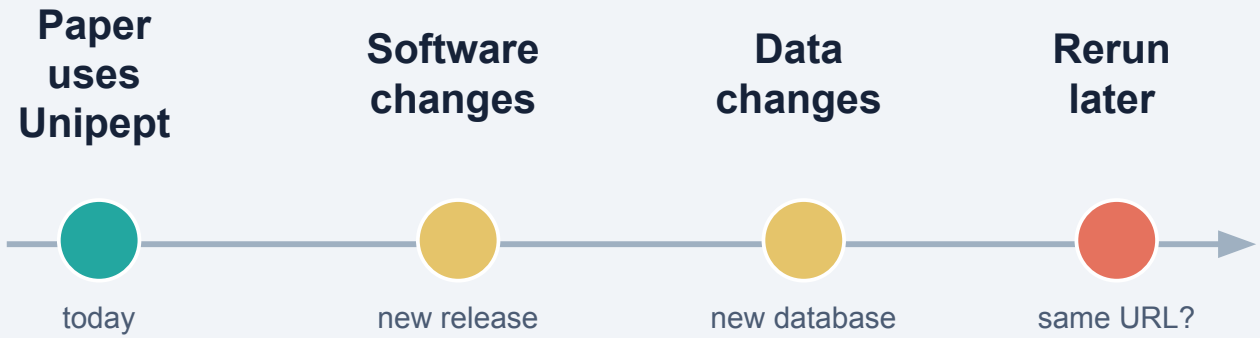
Reproducibility becomes harder

The same URL can point to a different result later, because the software and the underlying data keep moving.

What can change?

- **Software**
  Bug fixes, new features, changed defaults.
- **Reference data**
  In our case, UniProt is updated regularly.
- **Execution environment**
  Dependencies and infrastructure may drift too.

What that means in practice

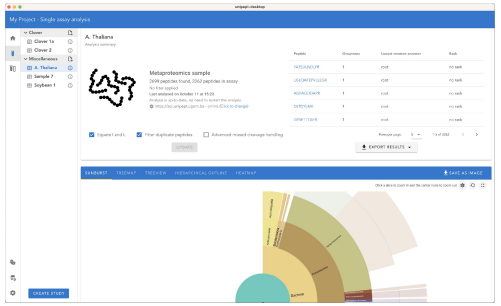


To rerun an old result, you need a frozen reference to the code, the data, and ideally the environment.

Our practical answer: a desktop app

Keep the convenience of the web, but also offer a packaged path that can be frozen and rerun later.

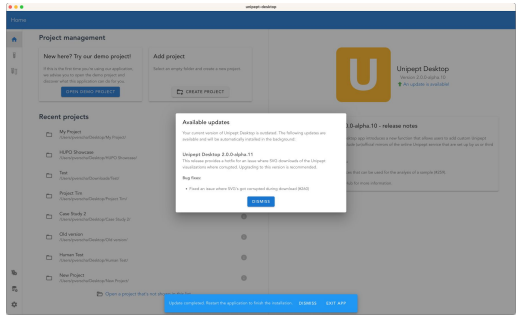
familiar interface



Unipept Desktop keeps the interface approachable using Electron.

local backend

Same analysis logic



Users can keep working with the same logic, but now with an option to run locally using Docker.

fixed state

Versioned app + versioned data

Gives the user the opportunity to set up an older state they can rerun.

reproducible results

Where to start

Follow general software engineering best practices

1	**Use Git + GitHub/GitLab**	History, collaboration, backup, issues and one stable home for the project.
2	**Provide documentation**	Explain what the project is, how to run it, and where the important files live.
3	**Make releases, not just deployments**	Tag versions and keep short release notes so old states are still identifiable.
4	**Add tests and CI**	Let a machine check that the project still works when code or dependencies change.
5	**Manage dependencies**	For exact reruns, pin the data snapshot, dependencies and execution environment too.

These tips will not only help with reproducibility, but will also improve maintainability and software quality.