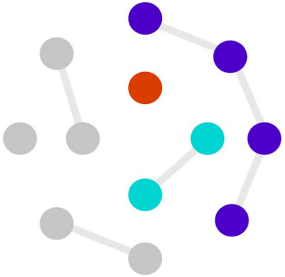
Garvan Institute
of Medical Research

Leveraging Continuous Integration for enhanced eResearch on High Performance Computing Clusters

Ignatius Menzies - Software Engineer; Data Science Platform

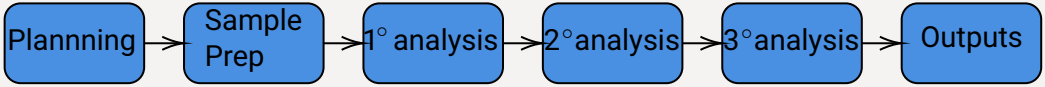
Garvan Institute
of Medical Research



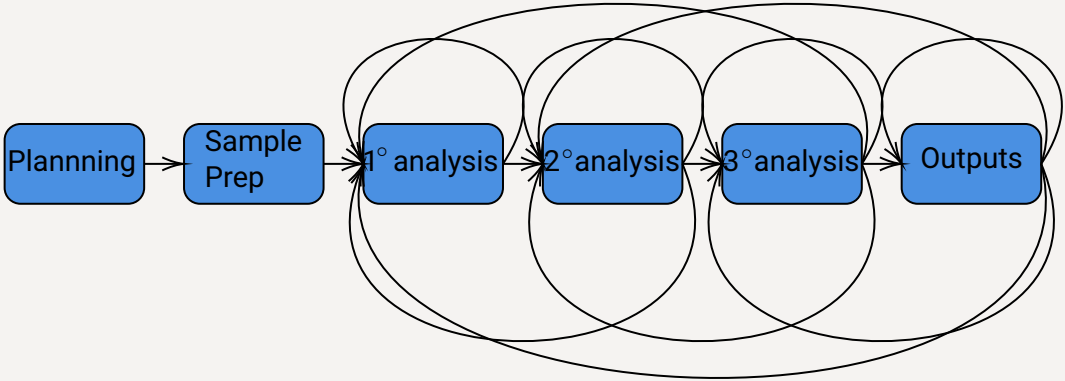
You can’t spell Reproducibility without CI

Research is ~~messy~~ iterative and non-linear

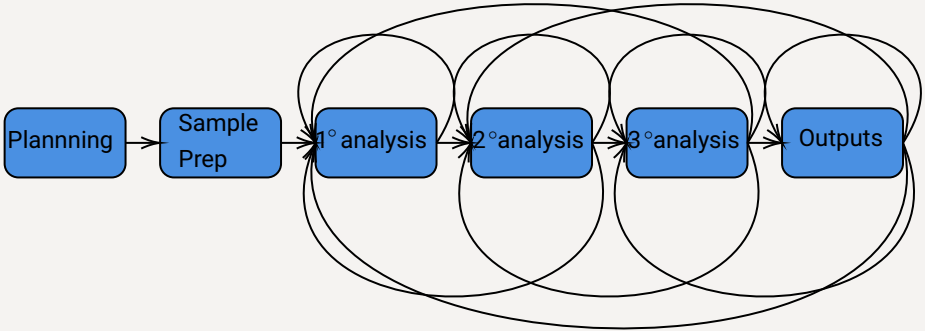
Research is ~~messy~~ iterative and non-linear



Research is ~~messy~~ iterative and non-linear

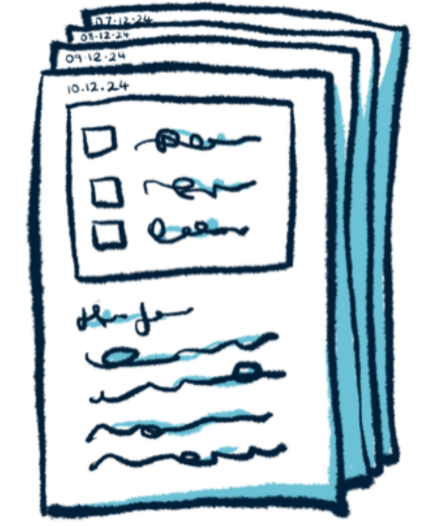


Research is ~~messy~~ iterative and non-linear

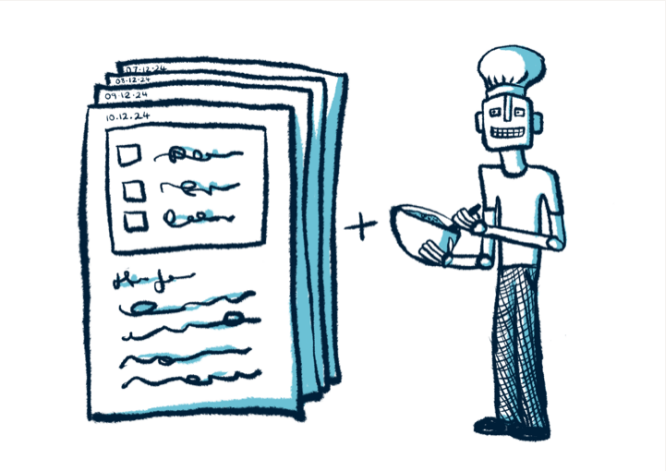


- High risk of errors
- Inefficient use of researcher time
- Not fun - can lead to shortcuts

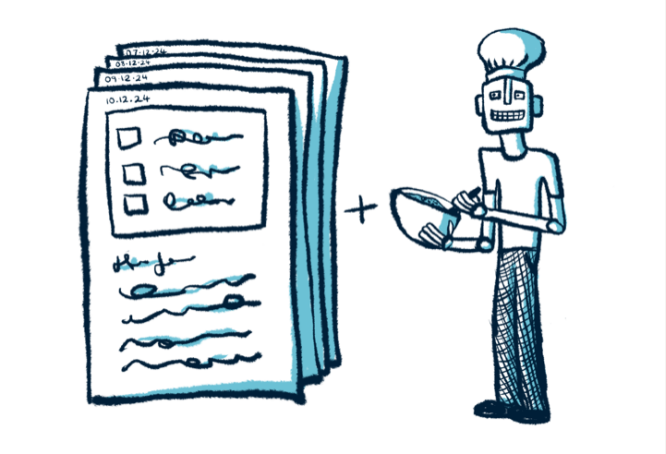
Reproducible research: a cooking analogy



Reproducible research: a cooking analogy

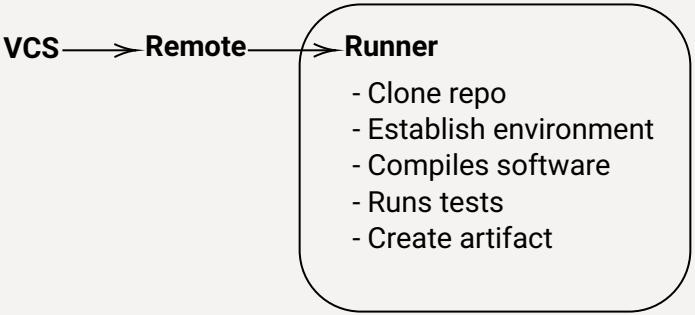


Reproducible research: a cooking analogy

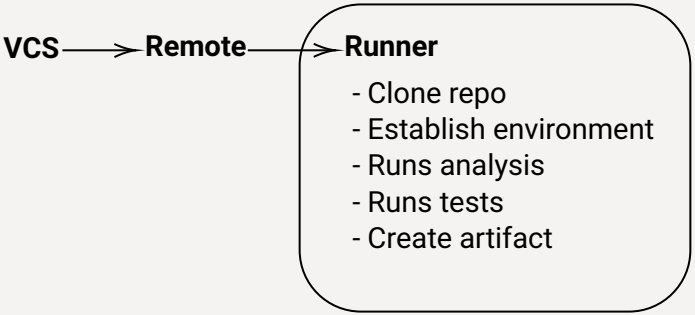


Continuous Integration (CI) is like a robot sous-chef for your research.

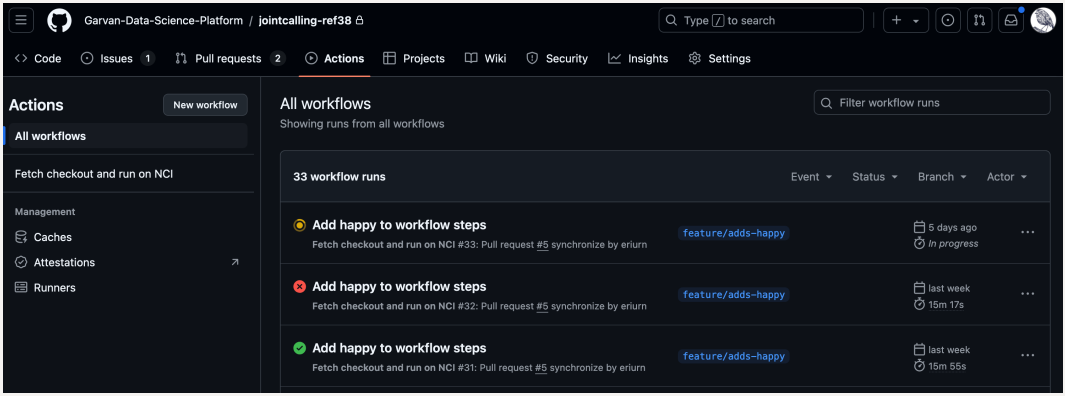
CI in software development



CI in Research



Pass or fail



Continuous Integration (CI) provides automated testing and rapid feedback.

But what about HPC?

Integrating CI and HPC is challenging:

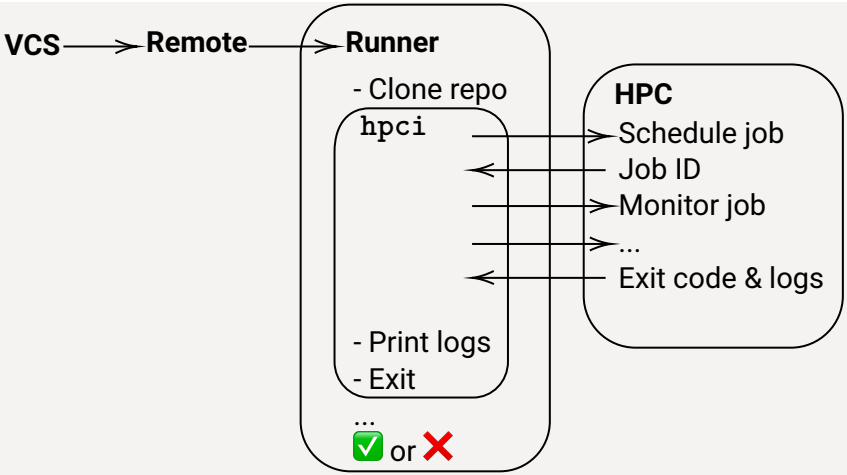
- Data is typically too large to be moved to runner
- CI runner has insufficient compute capacity
- Interface to HPC is typically interactive
- Scheduler adds indirection

Existing approaches for integrating CI and HPC

- Jacamar CI (*https://gitlab.com/ecp-ci/jacamar-ci*)
- Jenkins-CI with HPC (*https://journals.sagepub.com/doi/full/10.1177/1087057116679993*)
- HPC-Rocket (*https://github.com/SvenMarcus/hpc-rocketg*)

HPC + CI = `hpci`

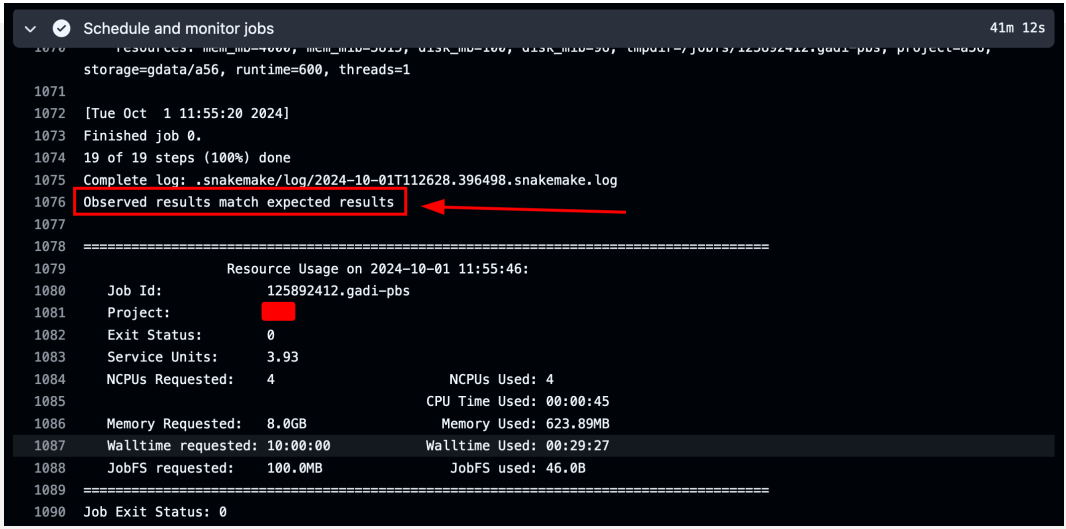
HPC + CI = `hpci`


VCS
Remote
Runner
- Clone repo
hpci
- Print logs
- Exit
...
or
HPC
Schedule job
Job ID
Monitor job
...
Exit code & logs

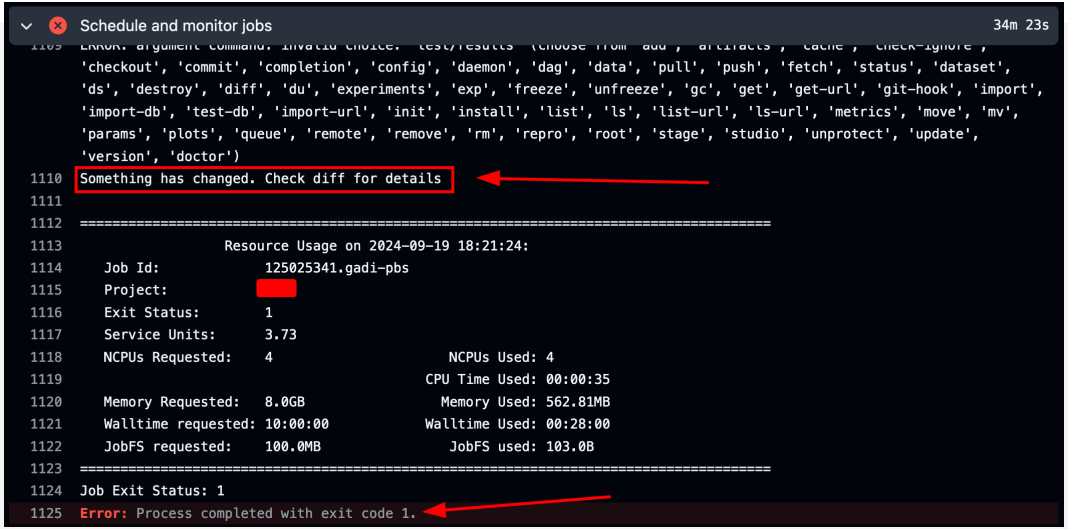
How we're using `hpci` at Garvan

`hpci` helps provide automated validation for bioinformatics workflows.

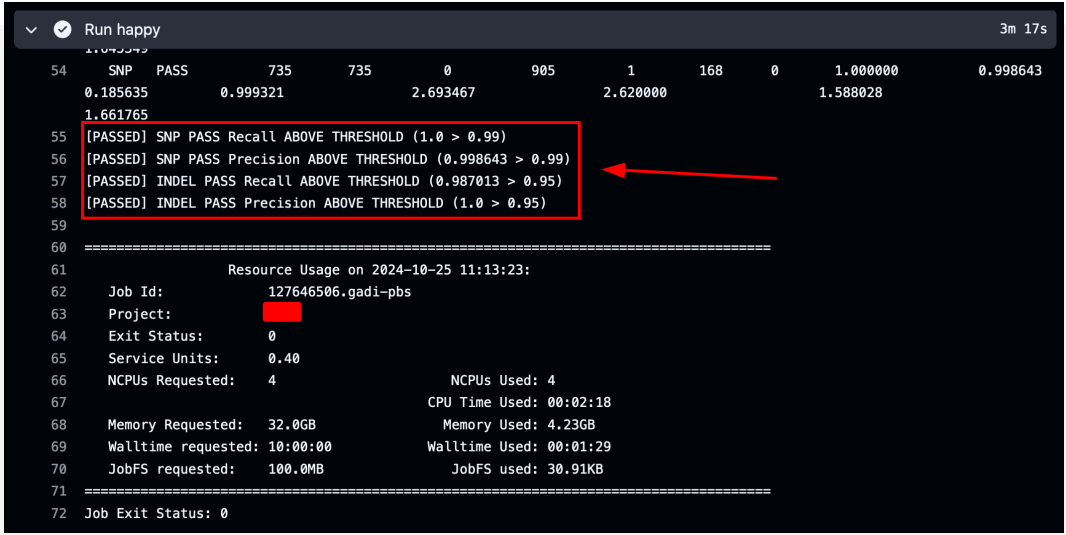
How we're using `hpci` at Garvan



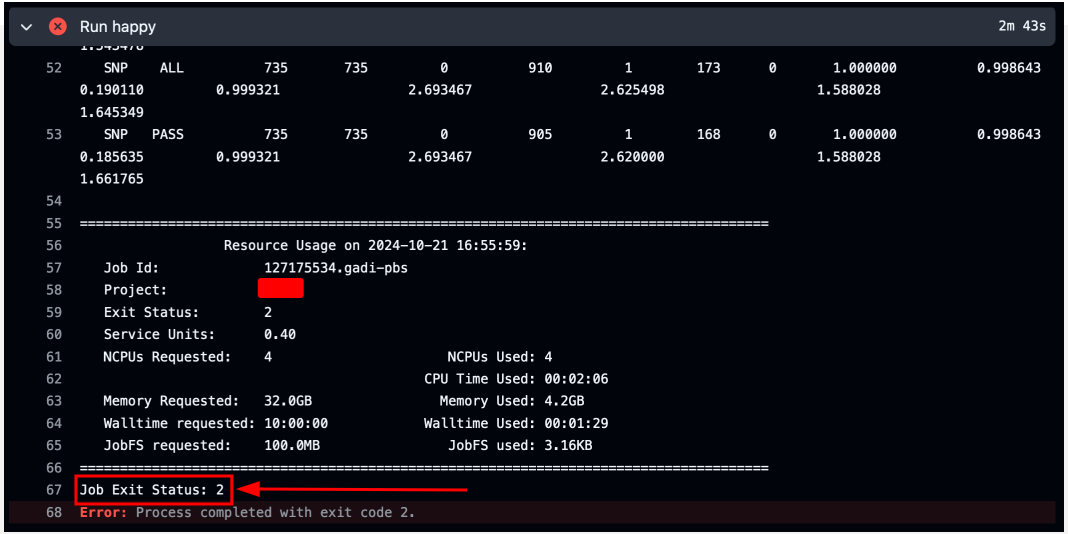
How we're using `hpci` at Garvan



How we're using `hpci` at Garvan



How we're using `hpci` at Garvan



HPC + CI = `hpci`

Automated testing reduces the likelihood of errors in production workflow runs.

The tests run on the same hardware as the production workflows, ensuring consistency.

Frequent, small validation jobs are more efficient than errors during much larger production jobs.

hpci

Huge thanks to my Garvan collaborators:

- John Reeves and Victor Liu from Cancer Ecosystems Program; and,
- Joe Copty and Eric Urng from DSP Production Bioinformatics.

Get in touch if you are interested in using `hpci`, or if you have any other ideas where `hpci` could be useful (*i.menzies@garvan.org.au*).

https://github.com/Garvan-Data-Science-Platform/hpci/