Making Bioinformatics Accessible

Andrew Wheeler

11/27/2024

Roots for Resilience Program

Arizona Institute for Resilience

RESEARCH, INNOVATION & IMPACT
Data Science Institute

CYVERSE®

- Fellowship and outreach program
- Build data science skills
- Connect people across disciplines

Open Science Principles

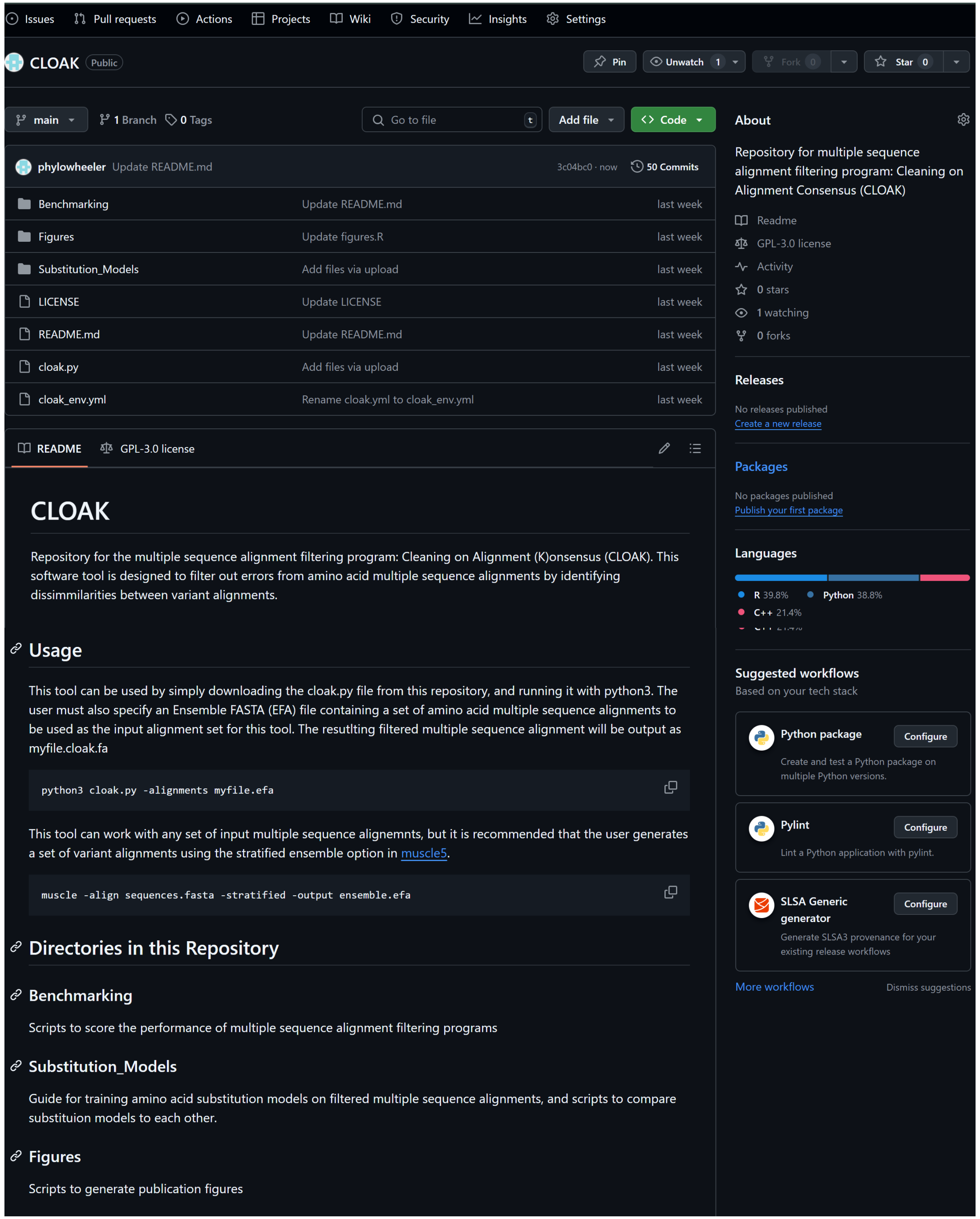
- Accessibility
  - Data management
  - Code availability
  - Documentation
- Reproducibility
  - Software Environments
  - Containers
  - Pipelines

Data Management

- Can users find and access your data?
- Can they understand, use, and reuse it consistently?
- Are you ethically protecting data where needed?
- Data management plans

GitHub

- Version controlled code development platform
  - Make code available to public
  - Track changes
  - Work with team
- README
  - Where to find everything
  - How to run code



phylowheeler Update README.md 3c04bc0 · now 50 Commits

Benchmarking	Update README.md	last week
Figures	Update figures.R	last week
Substitution_Models	Add files via upload	last week
LICENSE	Update LICENSE	last week
README.md	Update README.md	last week
cloak.py	Add files via upload	last week
cloak_env.yml	Rename cloak.yml to cloak_env.yml	last week

CLOAK

Repository for the multiple sequence alignment filtering program: Cleaning on Alignment (K)onsensus (CLOAK). This software tool is designed to filter out errors from amino acid multiple sequence alignments by identifying dissimmilarities between variant alignments.

Usage

This tool can be used by simply downloading the cloak.py file from this repository, and running it with python3. The user must also specify an Ensemble FASTA (EFA) file containing a set of amino acid multiple sequence alignments to be used as the input alignment set for this tool. The resultling filtered multiple sequence alignment will be output as myfile.cloak.fa

```
python3 cloak.py -alignments myfile.efa
```

This tool can work with any set of input multiple sequence alignemnts, but it is recommended that the user generates a set of variant alignments using the stratified ensemble option in muscle5.

```
muscle -align sequences.fasta -stratified -output ensemble.efa
```

Directories in this Repository

Benchmarking

Scripts to score the performance of multiple sequence alignment filtering programs

Substitution_Models

Guide for training amino acid substitution models on filtered multiple sequence alignments, and scripts to compare substituion models to each other.

Figures

Scripts to generate publication figures

About

Repository for multiple sequence alignment filtering program: Cleaning on Alignment Consensus (CLOAK)

Readme
GPL-3.0 license
Activity
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

● R 39.8% ● Python 38.8%
● C++ 21.4%

Suggested workflows

Based on your tech stack

Python package
Create and test a Python package on multiple Python versions.

Pylint
Lint a Python application with pylint.

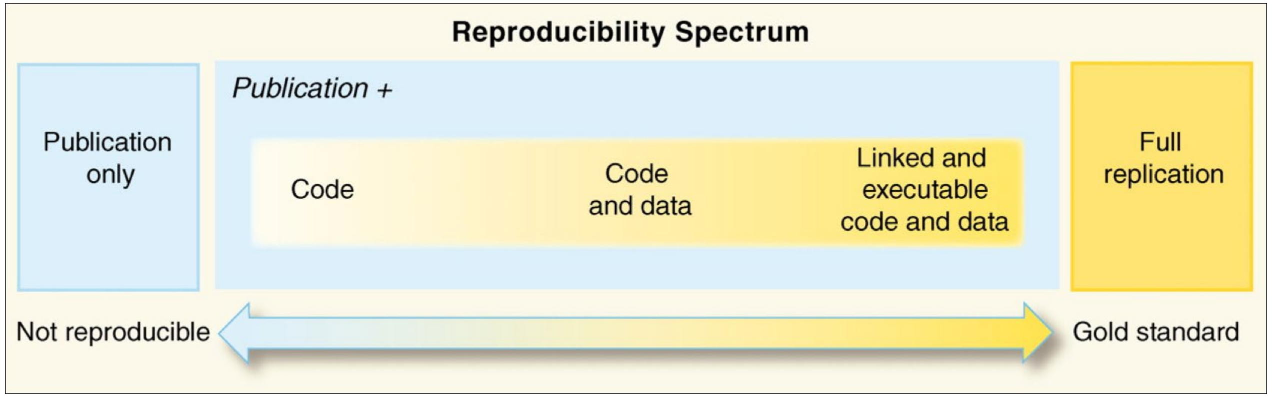
SLSA Generic generator
Generate SLSA3 provenance for your existing release workflows

GitHub

- Basic Commands
  - clone: copy a repository locally
  - pull: update local repository
  - branch: a version history. Multiple parallel branches can be active
  - fork: copy of someone else’s repository stored on your account
  - commit: finalize a change
  - push: add change back to remote repository
  - merge: apply changes from a branch or fork to the main branch
  - pull request: submit changes to be added t repository
  - issue: flag suggestions or tasks

Reproducibility

- Can users run your code?
- Will they get the same outcome as you if they do?
- Can users apply your code to their own projects easily?



Reproducibility Spectrum

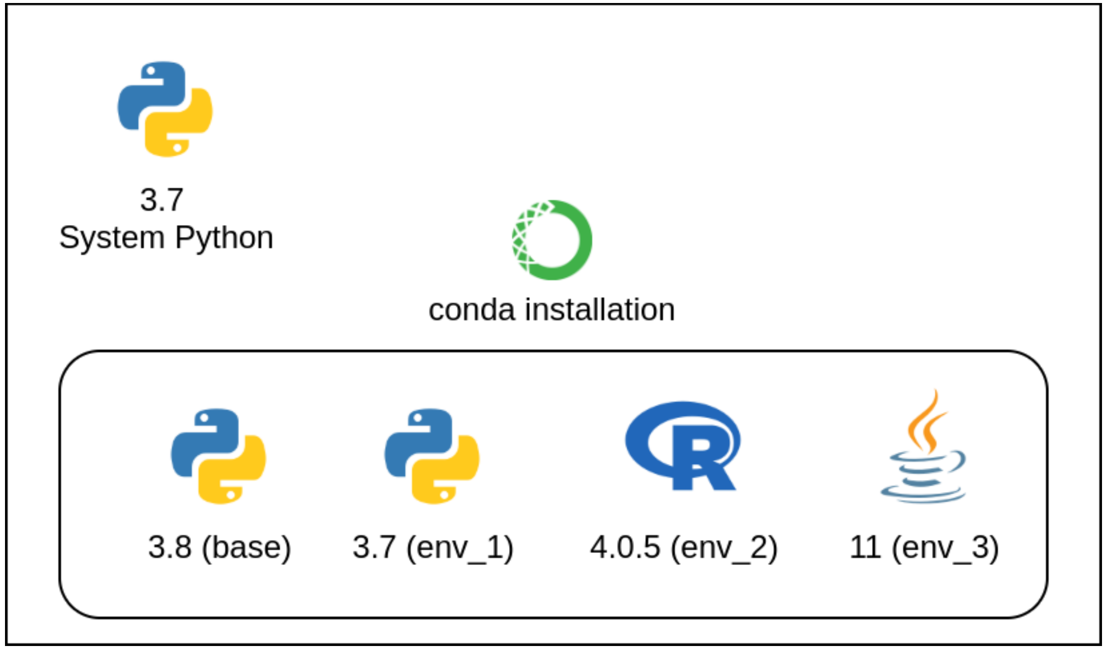
Computing Environments

- All the hardware, software, and resources you are using
    - Hardware: CPUs, GPUs, RAM
    - Operating System: Windows, Mac, Linux
    - Software Versions: R, Python, etc.
    - Packages and Package Versions: specific software packages
- Software Dependency Hell:
    - Incorrect versions
    - Missing dependencies
    - Obsolete code

Environment Managers

Environments can be exported and shared so all users have the same versions

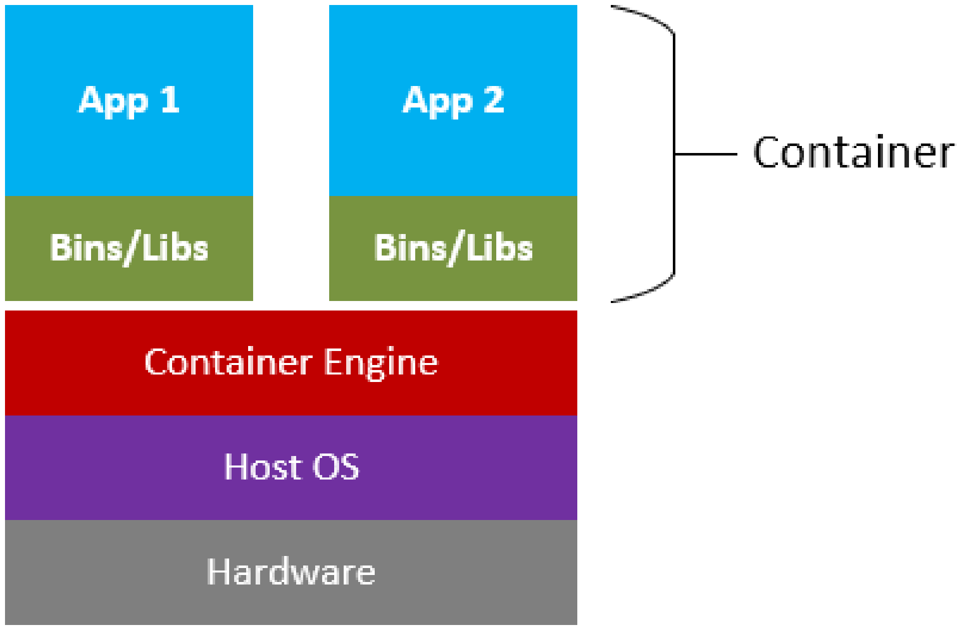
- Conda – most popular
- Mamba – implemented in C, faster than Conda
- Pip
- Renv



Export: conda env export > my_conda_env.yml
Reproduce: conda env create --file environment.yml

Containers

- Contains everything you need to run the code in a single unit
  - Easy to share
  - Can run on any machine
  - Isolated from the rest of the computer
- Docker – most popular container management software

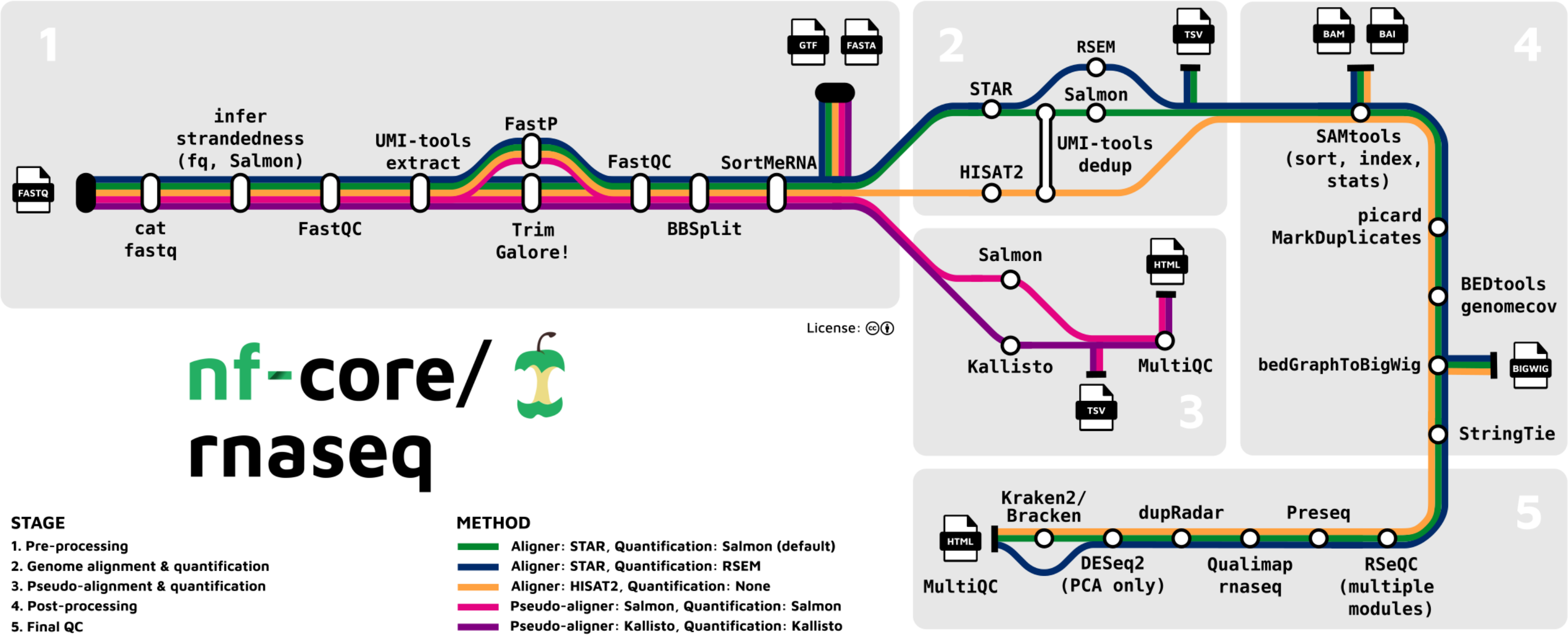


Containers

Pipelines

Workflow managers can let you standardize complex, multistep analyses

- NextFlow
- Snakemake

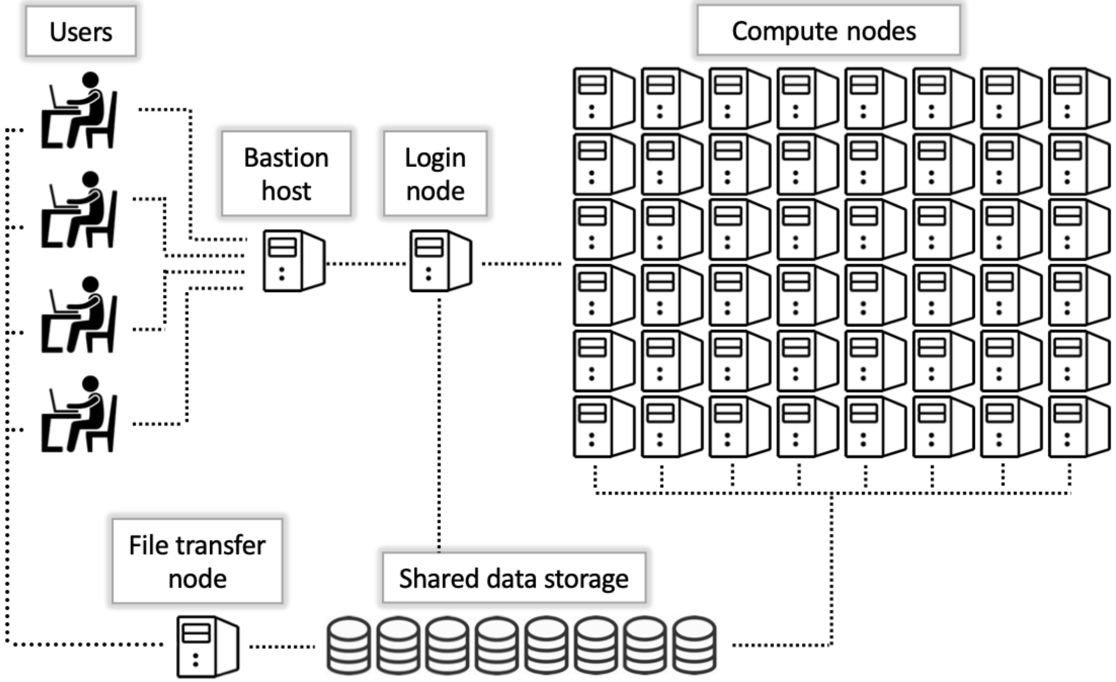


Remote Computing

Working beyond your local machine

- HPC – high performance computing
- CyVerse – cloud computing service hosted by UA

- Computing clusters
  - Large number of machines linked together
  - Power from parallelization



Summary

How can we make apply open science principles to bioinformatics?

- Well documented, easily accessible code and data on github or other cloud services
- Software environments to avoid dependency hell
- Containers and Pipeline managers to handle larger projects
- Making use of available remote computing resources

FOSS Course

Foundational Open Science Skills
https://foss.cyverse.org/

Consider attending in future terms if you want to learn more!

Thanks!