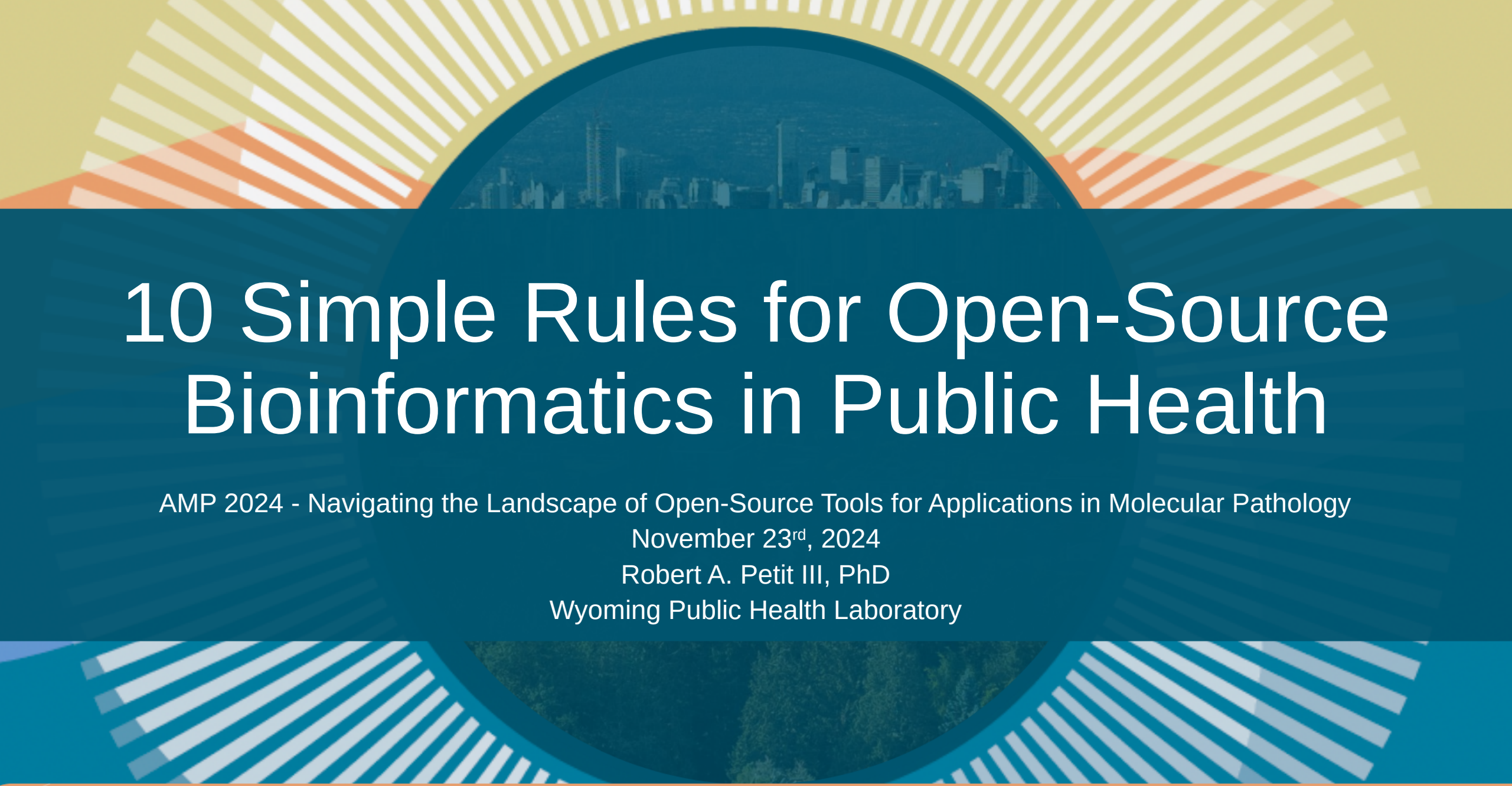


10 Simple Rules for Open-Source Bioinformatics in Public Health

AMP 2024 - Navigating the Landscape of Open-Source Tools for Applications in Molecular Pathology

November 23rd, 2024

Robert A. Petit III, PhD

Wyoming Public Health Laboratory

AMP Association for Molecular Pathology

PHOTOGRAPHY PERMITTED

Overview

- A little about myself and Wyoming
- A quick look at the last 15 years of bioinformatics
- 10 Simple Rules for Open-Source Bioinformatics in Public Health
- Wrap it all up

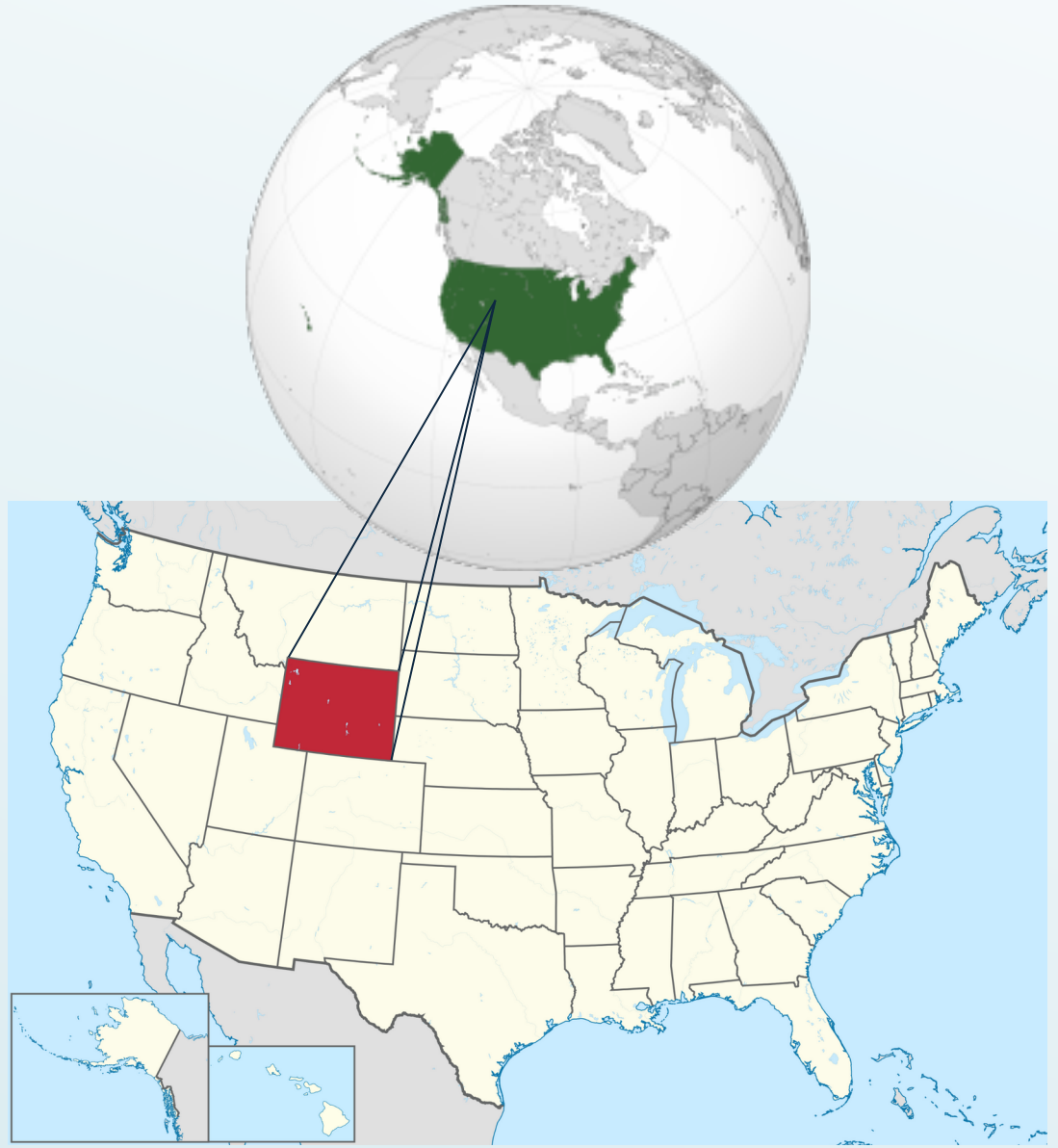


Howdy! 👋
I am Robert

Bioinformatician at the Wyoming Public Health Laboratory

Wyoming, USA

- Total Area 253k km^2
- Total Population ~580k
- Average elevation of 2,040 meters
- Many national parks filled with wildlife *(especially of the large sorts)*



Wyoming Public Health Laboratory (WPHL)

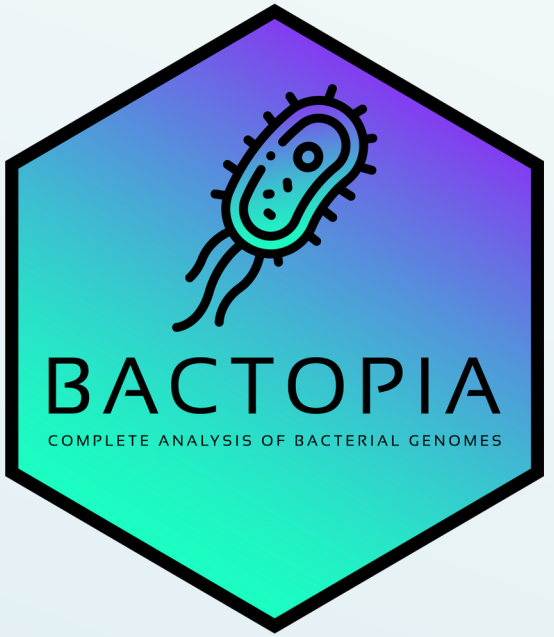
- Cheyenne, WY, USA
- 20-30 lab members
- Close relationship with:
  - State Vet Lab
  - State Agriculture Lab
  - Game and Fish
  - Dept of Environmental Quality
- Sequencing capacity includes several Illumina and Oxford Nanopore instruments



Bactopia for accessible bacterial analysis

A Nextflow pipeline for end-to-end analysis of bacterial genomes

- Wraps 150+ bioinformatic tools into a single pipeline
- Bactopia Tools provide 60+ additional workflows for more science
- Available from Docker, Singularity, and Conda
- Sustained for 5+ years with support from WPHL, Emory University, CAPE



Ask me for a sticker!

bactopia.github.io

Open-source contributions

- Hundreds of contributions back to the bioinformatics community
  - New or fixed Bioconda recipes (and lots of PR reviews)
  - New modules for nf-core/modules
  - Bug fixes and features to upstream tools

- Tools originally developed for Bactopia made standalone
  - camlhmp - Classification through yAML Heuristic Mapping Protocol
  - sccmec - A tool for typing SCCmec cassettes in assemblies
  - dragonflye - Assemble bacterial genomes from Nanopore reads
  - fastq-dl - Download FASTQ files from SRA or ENA repositories
  - Many more!

- Nearly 15 years of open-source development and contributions



Ask me for a sticker!

A lot has changed over the last 15 years

2010 and earlier
Bioinformatics is a bit chaotic and disconnected. Tools are hard to discover, let alone even install.

2013
While “containers” aren’t a new concept, Docker popularized their usage in bioinformatics

2014 -2017
Worflow managers start appearing in the wild. Including Nextflow, Snakemake and WDL.

2017
Staph-B and Bioconda are formed, bringing pre-built packages and containers

2020-2021
COVID-19 makes genomic epidemiology mainstream, bringing sequencing to many PHLs across the globe

Today
Sequencing is now a standard practice, but bioinformatic accessibility is still lagging



To help make new tools accessible, I try to follow 10 simple rules for open-source bioinformatics in public health

Rule 1 - Avoid Reinventing the Wheel

Rule 1 - Avoid Reinventing the Wheel

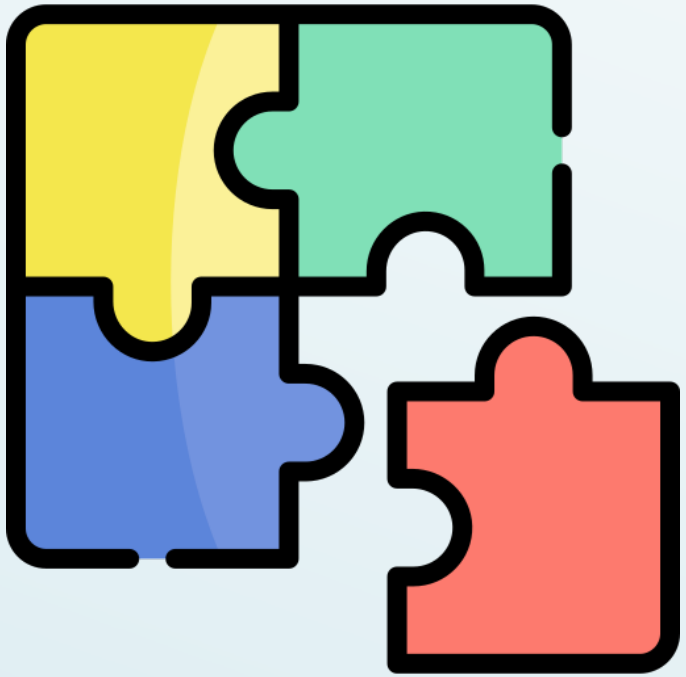
- Focus on your problem by avoiding the rebuilding process
- Save time and resources
- Existing community support
- Bugs are more likely to be resolved
- Ask yourself: Is there a tool already that solves my problem?



Rule 2 - Reinvent with Familiarity in Mind

Rule 2 - Reinvent with Familiarity in Mind

- When you must reinvent the wheel, work with what users already know
- Minimize the learning curve
- Aim to be a drop in replacement
  - Use similar interfaces, inputs, parameters, outputs, etc…
- Be compatible with community standards



Example: Prokka and Bakta

- Prokka was the go-to bacterial genome annotation tool for nearly 10 years
  - https://github.com/tseemann/prokka
- Bakta was released in 2020 as a drop-in replacement for Prokka
  - https://github.com/oschwengers/bakta
- Bakta maintained very similar inputs and outputs, but many additional types of annotation




Torsten Seemann @torstenseemann · Sep 1, 2022
Bakta is a worthy successor to Prokka, which has been neglected in recent years. Bakta is being actively developed with new features that Prokka never had, so I encourage Prokka users to try it out!
Oliver Schwengers @oschwengers1 · Sep 1, 2022
We've released v1.5.0 of Bakta along with a huge DB update!
New features:
- detection of pseudogenes
- assignments of KEGG Orthology accessions...
Show more
7
204
592

Rule 3 - Start Simple, Then Expand

Rule 3 - Start Simple, Then Expand

- Aim to solve the primary problem first
  - “minimum viable product”
- Make it available as a beta for user feedback
  - Improve based on feedback
- Avoid overengineering and overcomplicating
  - Save advanced features for later
  - Make it easy for yourself and users to make progress



Rule 4 - Start Documenting Yesterday

Rule 4 - Seriously, Document Something!

Rule 4 - Start Documenting Yesterday

- Start documenting early, when possible, try to document in the code
- Provide tutorials for users to get started
  - Test data
  - Output overview
- Details on how to report bugs and feedback
- Improve documentation overtime



Rule 4 - Document at least something!

- Make your documentation publicly available

- Utilize free resources
  - Frameworks
    - MkDocs
    - Read The Docs
  - Hosts
    - GitHub or GitLab Pages
    - Netlify
    - Read The Docs

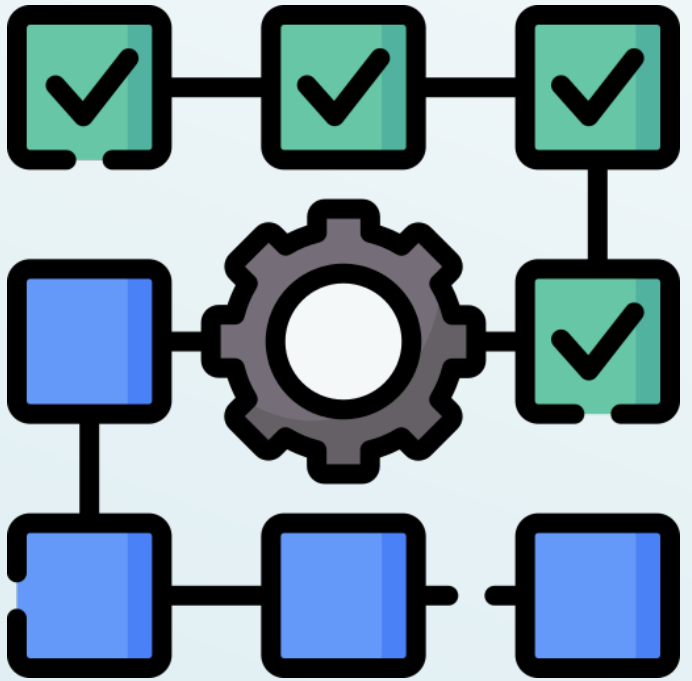
- Be creative, *have fun*, and remember the docs are also for your future self!



Rule 5 - Build It, Test It, Debug It

Rule 5 - Build It, Test It, Debug It

- Catch and fix issues early
- Implement continuous integration for automated testing
  - Many free resources (e.g. GitHub, GitLab)
- Use real data when possible
- Add edge cases as they get reported
  - Users are great at finding these!
- Document and report known issues
  - Is it a simple upstream fix?
  - Do you need to look at alternatives?





Celebrate the green check marks!

Rule 6 - Installation Shouldn’t Be a Barrier

Rule 6 - Installation Shouldn’t Be a Barrier

- Installation should be seamless and easy for users
- Simplify installation with package managers
- Bioconda
  - ~10,000 “bio” related tools
  - Docker and Singularity containers made available
  - *conda install some-tool*
- Remember Rule 4 Documentation? Detail installation there!



Rule 7 - Let Workflow Managers Do the Heavy Lifting

Rule 7 - Let Workflow Managers Do the Heavy Lifting

- Workflow managers provide:
  - Accessibility
  - Standardization
  - Portability
  - Reproducibility
  - Resilience
  - Scalability

- There are numerous workflow managers available. Remember Rule 1?

Common Workflow Managers

nextflow

snakemake

{wdl}

Galaxy PROJECT

```
[skipped  ] process > BACTOPIATOOLS:DEFENSEFINDER:DEFENSEFINDER_UPDATE (update)          [100%] 1 of 1, stored: 1 ✓
[6d/12d9ac] process > BACTOPIATOOLS:DEFENSEFINDER:DEFENSEFINDER_RUN (SAMN31104671)       [100%] 644788 of 644788, failed: 89, retries: 89 ✓
[6d/9dbd16] process > BACTOPIATOOLS:DEFENSEFINDER:GENES_CONCAT (defensefinder-genes)     [100%] 1 of 1 ✓
[27/beb48b] process > BACTOPIATOOLS:DEFENSEFINDER:HMMER_CONCAT (defensefinder-hmmer)     [100%] 2 of 2, failed: 1, retries: 1 ✓
[9a/3dd68c] process > BACTOPIATOOLS:DEFENSEFINDER:SYSTEMS_CONCAT (defensefinder-systems) [100%] 3 of 3, failed: 2, retries: 2 ✓
[01/410378] process > BACTOPIATOOLS:CUSTOM_DUMPSOFTWAREVERSIONS (1)                     [100%] 1 of 1 ✓

    Bactopia Tools Execution Summary
    ---------------------------------
    Workflow         : defensefinder
    Bactopia Version : 3.1.1
    Nextflow Version : 24.04.4
    Command Line     : nextflow run /home/rpetit/.conda/envs/bactopia-dev//share/bactopia-3.1.1/main.nf -w /gscratch/rpetit/atb/atb-assemblie
    Profile          : arcc
    Resumed          : false
    Completed At     : 2024-10-31T19:22:06.055328084-06:00
    Duration         : 6d 3h 59m 33s
    Success          : true
    Exit Code        : 0
    Error Report     : -
    Launch Dir       : /pfs/tc1/gscratch/rpetit/atb/atb-assemblies/salmonella_enterica
    Merged Results   : bactopia//bactopia-runs/defensefinder-20241025-152333
```

Bactopia letting Nextflow do the heavy lifting while processing ~650k *S. enterica* assemblies from the AllTheBacteria Project

Rule 8 - Get to know your users

Rule 8 - Get to know your users

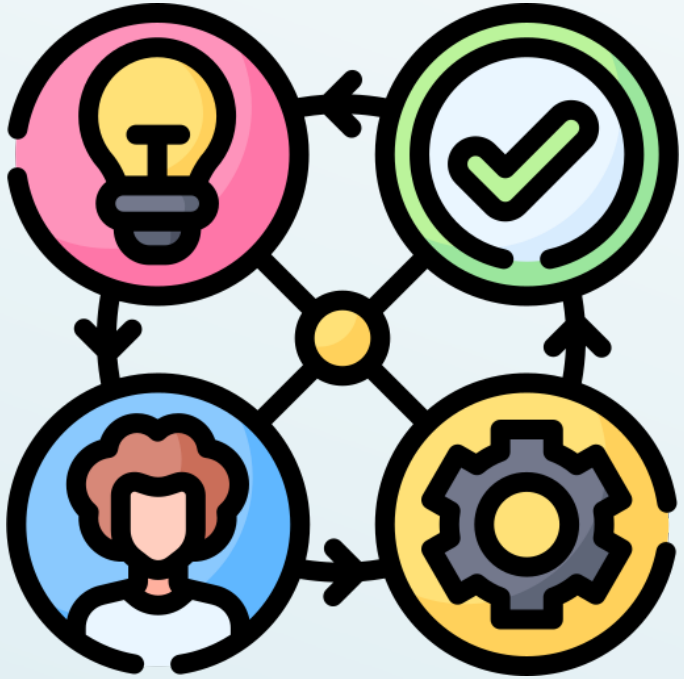
- Users often know better what's needed!
  - Challenges in usage, missing features, bugs…
- Encourage feedback and *actively listen*
  - GitHub issues, surveys, social media
- Build a community and collaborations
- Tailor to your users
  - Tutorials, case studies, blog posts



Rule 9 - Be your own power user

Rule 9 - Be your own power user

- “Eating your own dog food”
- Identify pain points for users, before they do
- Drive iterative improvements and discover new features
- Improve on your past self’s implementation



Rule 10 - Plan for Updates and Maintenance

Rule 10 - Plan for Updates and Maintenance

- A major issue in the bioinformatics field
- The field of bioinformatics isn’t setup for maintaining tools
  - Limited funding for “maintenance”
  - Short life-span for most tools
- Aim to reduce the long-term maintenance burden now



How do I reduce my maintenance burden?

How do I reduce my maintenance burden? (while also wrapping this talk up!?)

- Rule 1 - Avoid reinventing the wheel
- Rule 2 - Reinvent with Familiarity in Mind
- Rule 3 - Start Simple, Then Expand
- Rule 4 - Start Documenting Yesterday!
- Rule 5 - Build It, Test It, Debug It
- Rule 6 - Installation Shouldn’t Be a Barrier
- Rule 7 - Let Workflow Managers Do the Heavy Lifting
- Rule 8 - Get to know your users
- Rule 9 - Be your own power user

Acknowledgements

The many developers of open-source software and the users of that are regularly providing feedback.

WPHL

Chayse Rowley & Taylor Fearing
Jim Mildenberger
Rob Christensen
Joseph Reed

Emory

Tim Read

EMERGENT Group

Ahmed Babiker

Shannon Petit

Find me for a sticker!

A final bonus rule…

Rules are meant to be broken

Figure out what works for you and your situation, stick with it, grow with it, but be open to changing things up in the future!

Contact Information | robert.petit@wyo.gov | GitHub: @rpetit3