Bioinformatics Tools, Environments, and Installations -3 hr

- APHL Consultants to create resources

- Objectives of this module and practical:
    - Present principles of environments (conda/micromamba, venv, exporting to $PATH)
    - Present installation from the command line, troubleshooting installation
    - Ensure trainees apply their prior learning of Linux permissions to ensure installations are executable
    - Presentation component will introduce common bioinformatics tools used later in the training
    - Practical component will install and export to $PATH the following on trainees' systems:
        - § Conda (or micromamba)
        - § Fastqc, multiqc
        - § Kraken2
        - § Singularity
        - § Nextflow
        - § Sra-toolkit
        - § Samtools
        - § Bcl2fastq
        - § Dorado
        - § Nextstrain
        - § Nanoplot[AB1]

Containers and registries 1 hr

- APHL Consultants to create resources

- Objectives of this module and practical:
    - Introduce concepts of images, containers, and registries as a solution to complexities of installations and environments in prior presentation/practical
    - Describe Docker, Podman, ECR, Singularity and demonstrate pulling images and running containers
    - Practical component: Install Docker CLI, pull and install MIRA-NF

Computer Environments

Funding Disclosure

- This work is supported by the United States Centers for Disease Control and Prevention funding under the Global Health Security Partnerships: Expanding and Improving Public Health Laboratory Strategies and Systems (Grant Number: NU2HGH000080).

Introduction to Containers

Virtualization

Containerization is a way to allocate resources on a system in a very compartmentalized way

- Virtual machines
  - Hypervisor
  - Maintaining an environment that partitions a host machine
- Containers
  - Individual software
  - Dependencies included
  - Reproducible environment
  - “Works on my machine”

Containerization Software

Different containerization software exists to run containers

- **Docker**
- **Apptainer (Formerly Singularity)**
- **ECR**
- **Podman**

There are advantages and disadvantages to the different container software

Root privileges

Running Docker requires root access

Depending on the IT restrictions available at your site, access to certain container software may be limited

Root privileges are not required by Apptainer, and Podman

User Groups

```
docker run --rm=True -v $PWD:/data -u $(id -u):$(id -g) staphb/<name-of-docker-image>:<tag>
<command> <--flags --go --here>
```

`-u $(id -u):$(id -g)`

By default, when Docker containers are run, they are run as the root user. This can be problematic because any files created from within the container will have root permissions/ownership and the local user will not be able to do much with them. The -u flag sets the container's user and group based on the user and group from the local machine, resulting in the correct file ownership.

Volumes

```
docker run --rm=True -v $PWD:/data -u $(id -u):$(id -g) staphb/<name-of-docker-image>:<tag>
<command> <--flags --go --here>
```

`-v $PWD:/data`

By default, containers are isolated environments. Without ensuring that there is a way to continue to access the data on the host system, the data inside will disappear when the container does. If the data is exported to std out, it will remain there, but the data can also be captured in a volume which is mounted from the home system ($PWD, current working directory) onto the container (/data, a folder within the container). Data which is generated in or moved to the volume will persist on the host system even after the container has been deleted.

Running containers

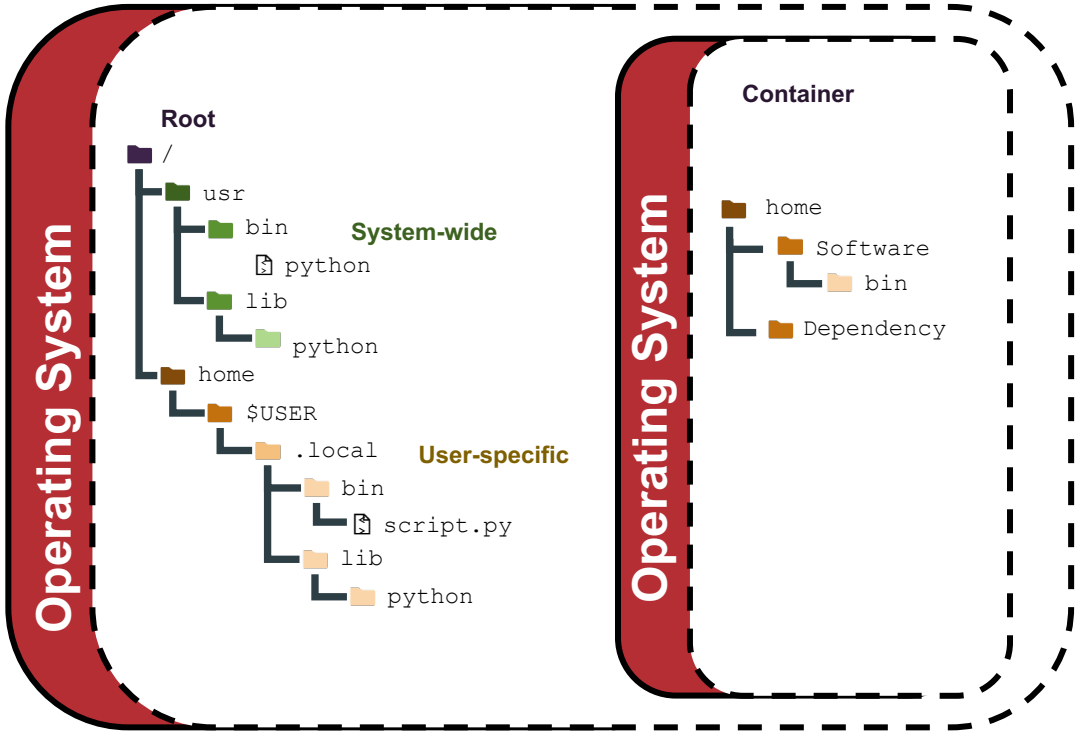
Containers can be run in interactive mode

- The flag for this is: -it
    - The i stands for interactive
    - The t stands for a pseudo TTY, that offers basic input/output

Containers can be called and given a specific command to run

Removal of containers that are no longer running saves space your system, but persistent containers can be called and used again, and any changes that were made during that session will remain

OS differences

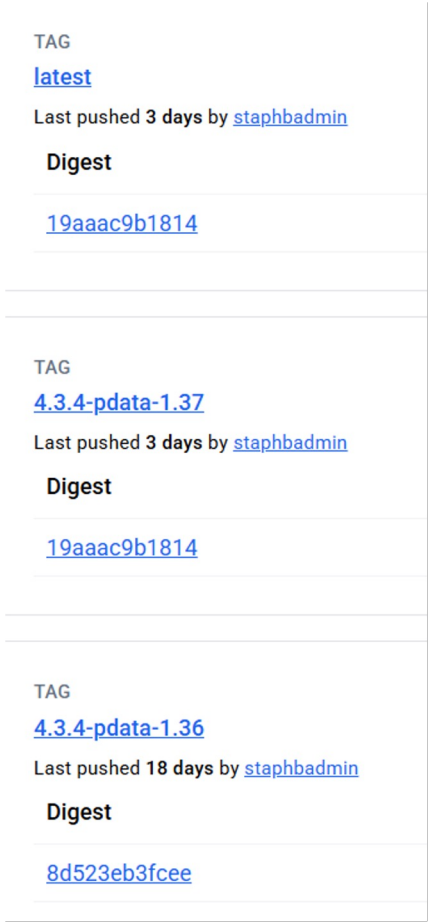


Containerization allows users to specify whatever operating system they would like to use within the container, whether that matches the host system or not

Container Versioning

Versions are specified in container naming/tags as a convention for noting which version of a software is in that container

“Latest” tag is less useful than more specific semantic versioning of a container, i.e. 1.1.0 since this will be more easily trackable and revertable (“latest” must be assigned, and authors do not always keep track or update the latest track, which can lead to problems downstream)



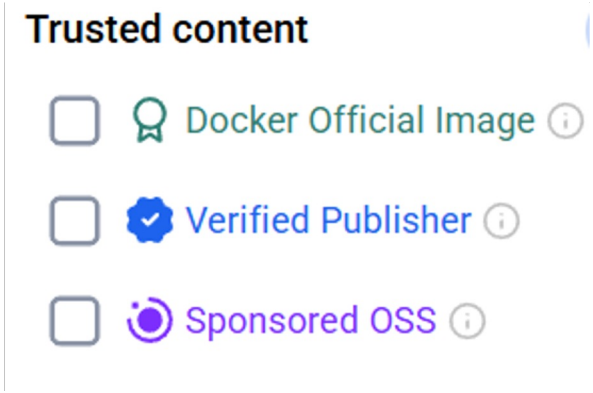
Repositories

[Dockerhub]

[Quay.io]

Using trusted sources of containers

- Reputable institutions (StaPH-B)



Activity

Download a Docker container

Run a container that has a specific software in it

Run a container in interactive mode

Run a container without a volume mounted

Common Bioinformatic Tools

Overview

We will be installing and using multiple common bioinformatics tools:

- Fastqc, multiqc
- Kraken2
- Apptainer (formerly Singularity)
- Nextflow
- Sra-toolkit
- Samtools
- Bcl2fastq
- Dorado
- Nextstrain
- Nanoplot

Bcl2fastq, Dorado

Bcl2fastq

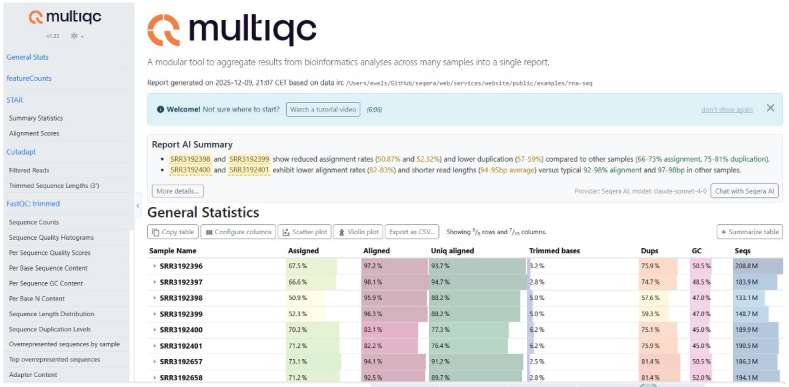
- Illumina software for creating fastq files from bcl files (clustering photos to sequence)
- Demultiplexing

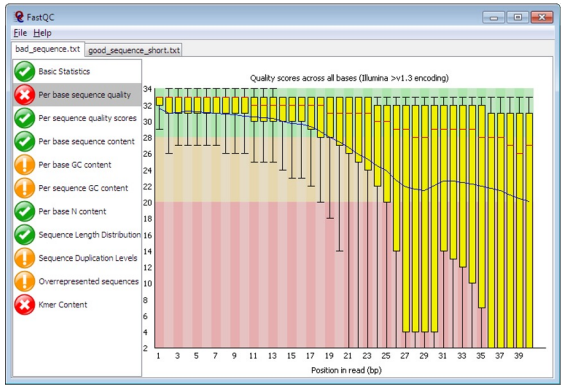
Dorado

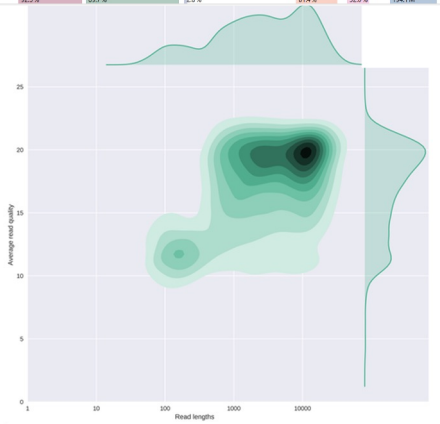
- Software for basecalling ONT reads from fast5 files
- Demultiplexing, alignment, trimming, correction, polishing

FastQC, MultiQC, and Nanoplot

- Quality control softwares that allow for viewing of sequencing data associated with sample quality
- Aggregate Q scores across reads
- Expected distribution of bases, visualizing adapter removal
- Visual representations of sequencing quality



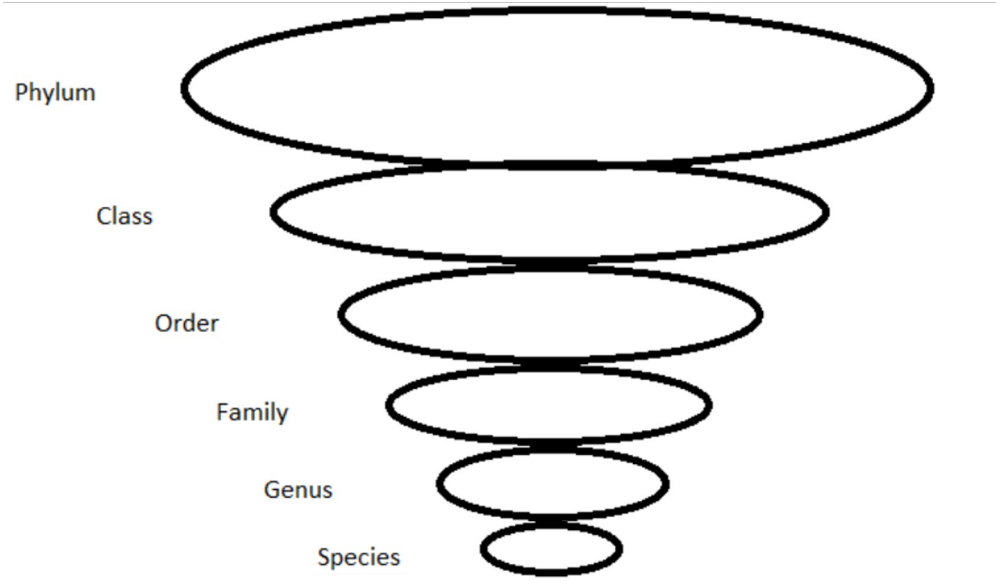




FastQC, MultiQC, Nanoplot

Kraken2

- Used for classifying reads to the most specific level of certainty within a taxonomic ranking scheme
- Dependent on a database
- Manual



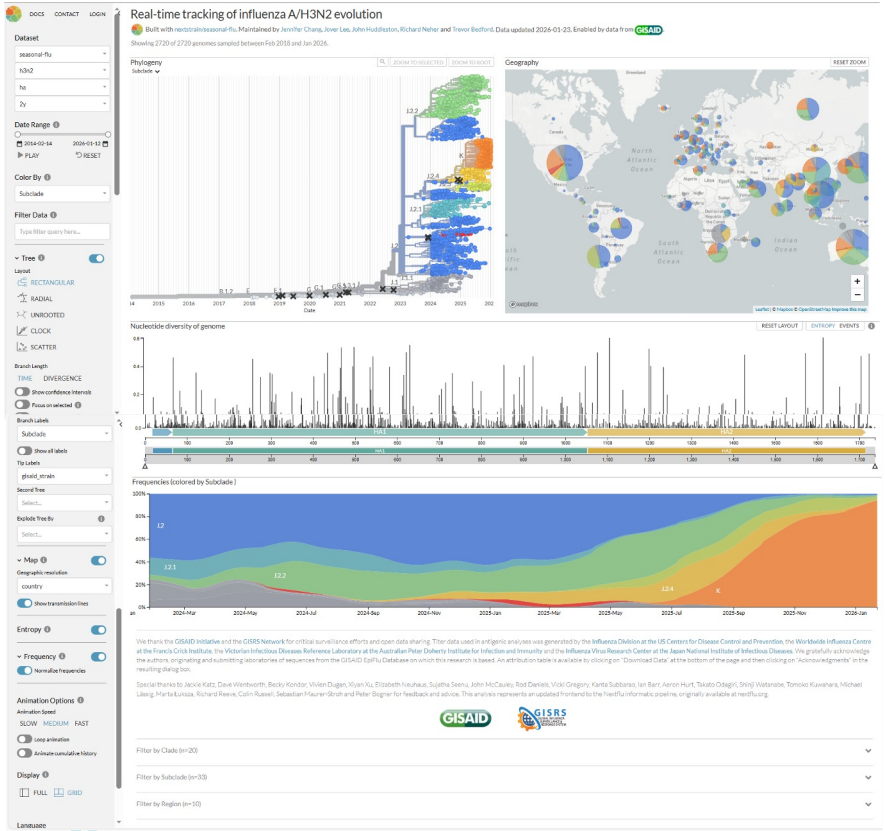
SRA-ToolKit

- Tools and libraries for using data in the INSDC Sequence Read Archives (SRA)
- Allows for downloading of NCBI data easily using command line
- SRA Accession numbers can be used to download data
  - Basic commands
    - prefetch
    - Fasterq-dump
- Manual

Samtools

- A set of tools for analyzing and working with sequencing data
- Samtools
  - Reading, writing, indexing, viewing, editing and working with files in SAM, BAM, CRAM format
- BCFTools
  - Reading, writing, BCF2, VCF, gVCF files for filtering and calling SNPs and short indels
- HTSlib
  - A C library for reading and writing high-throughput sequencing data
- Manual

Nextstrain



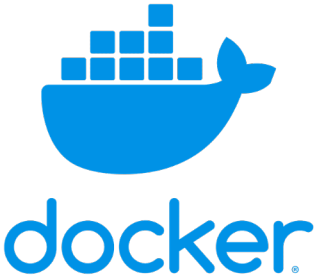
- A set of tools to create phylogenies and visualize genomic sequencing data
- Augur
  - Filter, align, tree build, refine, export
- Auspice
  - Visualization program for viewing trees and metadata overlays
- Can build a visualization with in-house data or use a preconfigured build

Nextflow

- Software for running containers in orchestrated ways, developing pipelines that optimize parallelization and create faster and reproducible ways to generate data
- Modular for ease of updating and changing
- Workflow tracking for troubleshooting and quality assurance
- Configurability
  - Config files for pipeline variable maintenance
- There exists a set of best-practice values for coding with, using, and contributing to nextflow called nf-core

Docker, Apptainer, Podman

- Softwares that run Docker containers
- Based on the security restrictions, one container may be a better fit for a user's organization



APHL Global Health webpage

Questions?

This work is supported by the United States Centers for Disease Control and Prevention funding under the Global Health Security Partnerships: Expanding and Improving Public Health Laboratory Strategies and Systems (Grant Number: NU2HGH000080).