

# Viral-NGS: Cloud Compute for Viral Genomics Using GA4GH Standards

Danny Park<sup>1</sup>, Chris Tomkins-Tinch<sup>1</sup>, Ilya Shlyakhter<sup>1</sup>, Christine Loreth<sup>1</sup>, Sushma Chaluvadi<sup>1</sup>, Simon Ye<sup>1</sup>, Lydia Krasilnikova<sup>1</sup>, Hayden Metsky<sup>1</sup>, Irwin Jungreis<sup>1</sup>, Mike Lin<sup>2</sup>, Pardis C. Sabeti<sup>1</sup>  
<sup>1</sup>Broad Institute of Harvard and MIT, Cambridge, MA, <sup>2</sup>DNAxenus & Chan Zuckerberg Institute, Mountain View, CA.



With support from

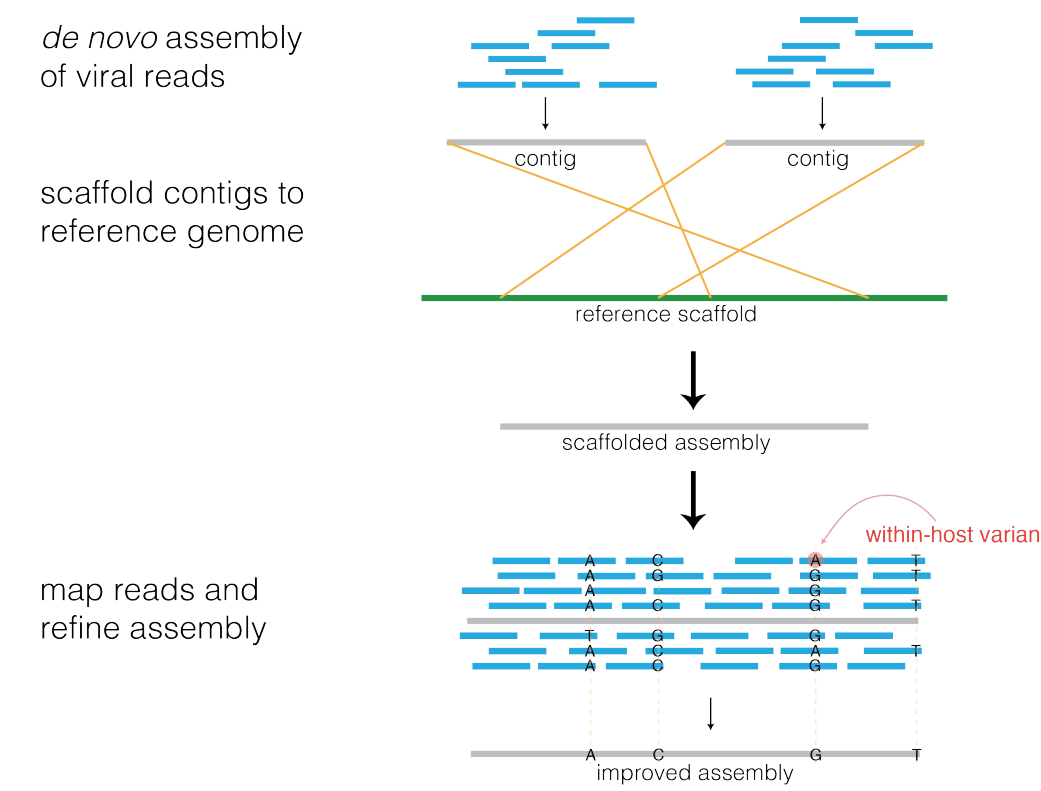


## The problem

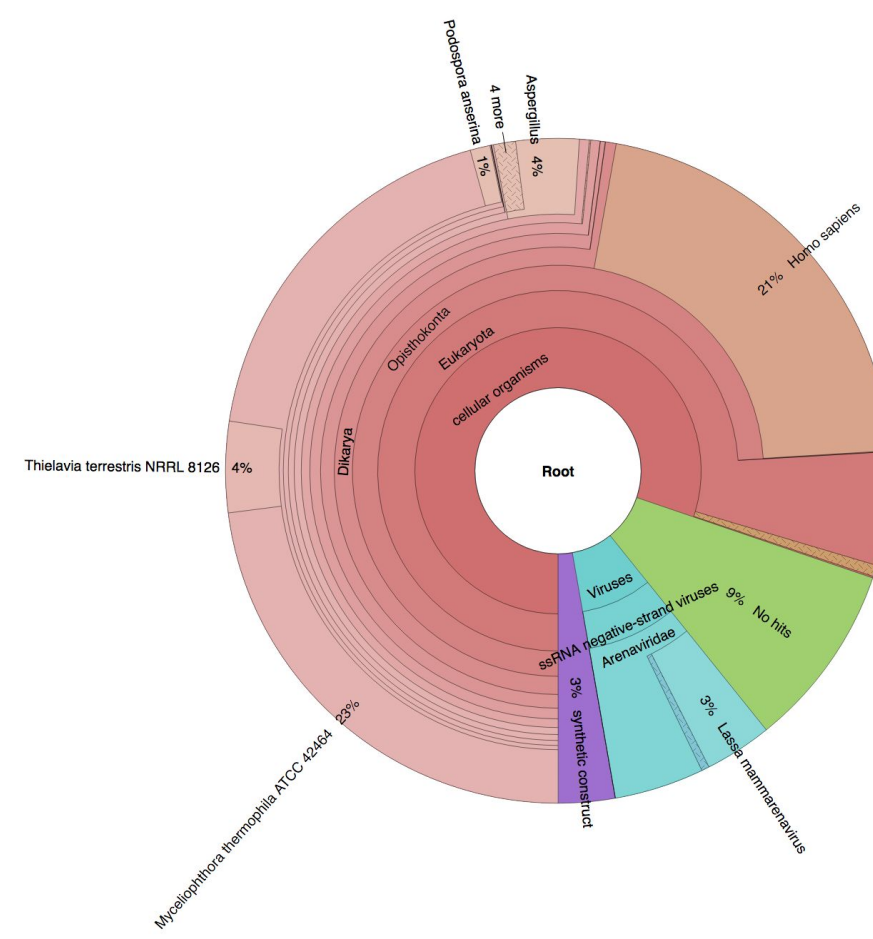
A long-standing challenge for bioinformatics has been to publish software tools that **prioritize portability across compute environments**, and **increase accessibility to a wide range of researchers**, enabling genomic analyses conducted directly by the labs and scientists that are producing the data and samples, all within a data environment they fully control and own.

## What it does

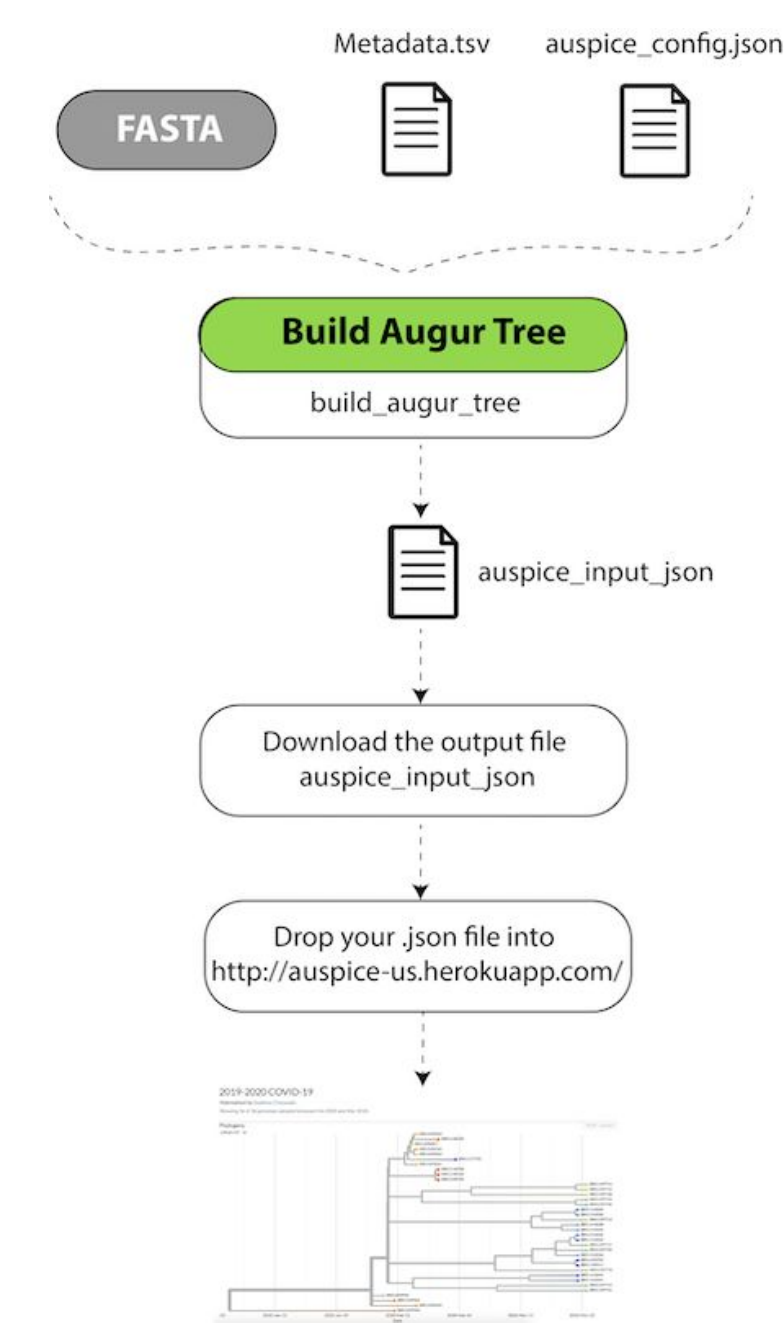
Genome assembly (de novo and reference based)



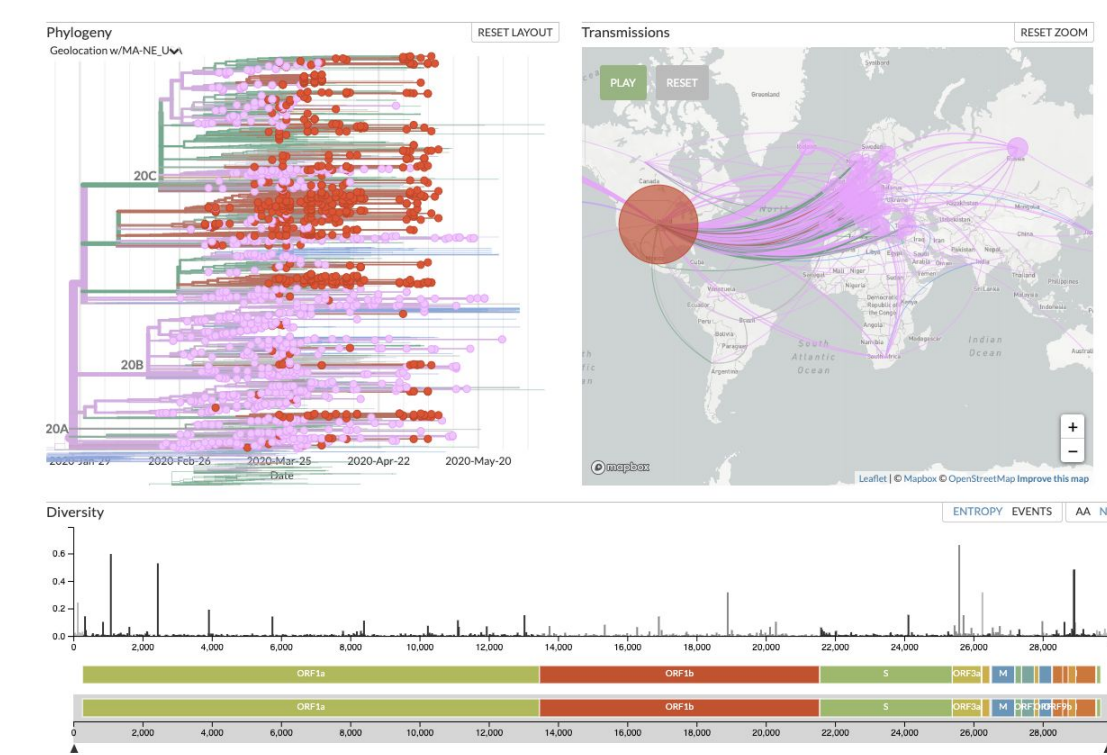
Metagenomics



Workspaces to manage data and compute (workflow & notebook style)



Phylogenetics & Nextstrain-based viz



Multi-platform supported



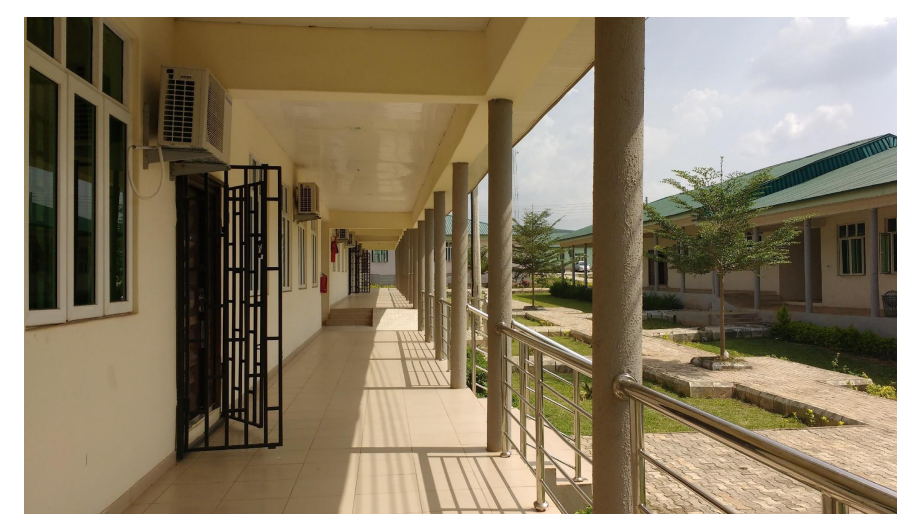
## Who uses it



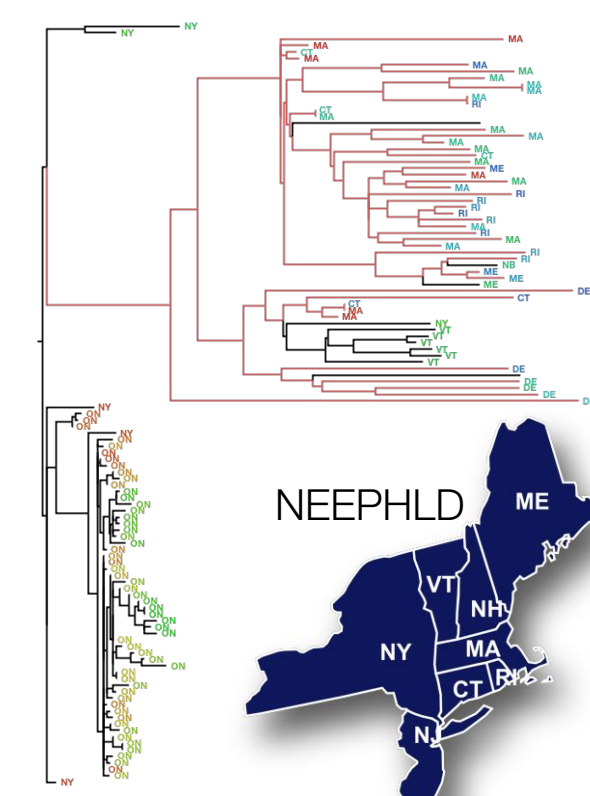
In our experience, cloud compute platforms facilitate much **faster adoption of the analysis work** by local research staff with less informatic resources or experience.

Enabled local investigations:

- 2014
  - Nigeria: Ebola
- 2015
  - Nigeria: unknown VHF death in hospital
- 2017
  - Nigeria: monkeypox
  - Sierra Leone: Ebola HCW retrospective
  - Senegal: non-malaria fevers, Dengue
- 2018
  - Nigeria: Lassa, Yellow fever
  - Senegal: antimalarial resistance
- 2020
  - Nigeria & Sierra Leone: SARS-CoV-2



Folarin, et al., *J Infectious Diseases* (2016)  
 Sidle, et al., *NEJM* (2018)



Enabled US State Public Health Lab NGS trainings:

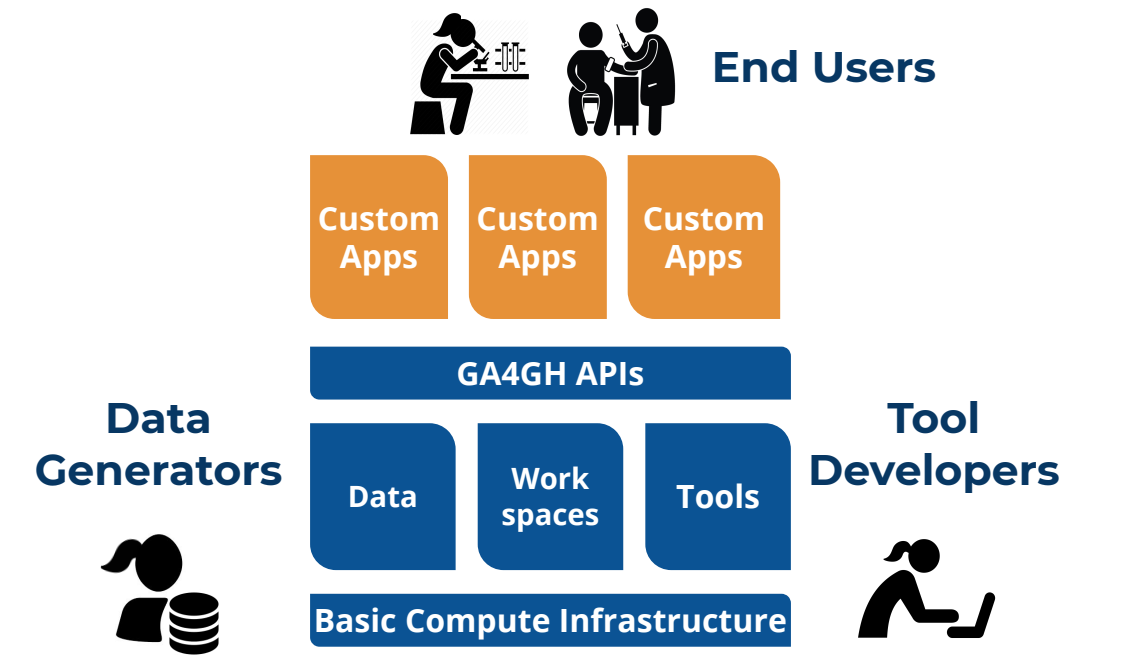
- Training SPHLs in MA, NH, VT, RI, CT, NJ, NY, DE since 2017
- Viral sequencing (lab) and viral metagenomics, assembly, and phylogenetics (bioinformatics)
- w/CDC OAMD & MA DPH

## How we achieve cross platform portability

Aggressively adopt existing interop standards

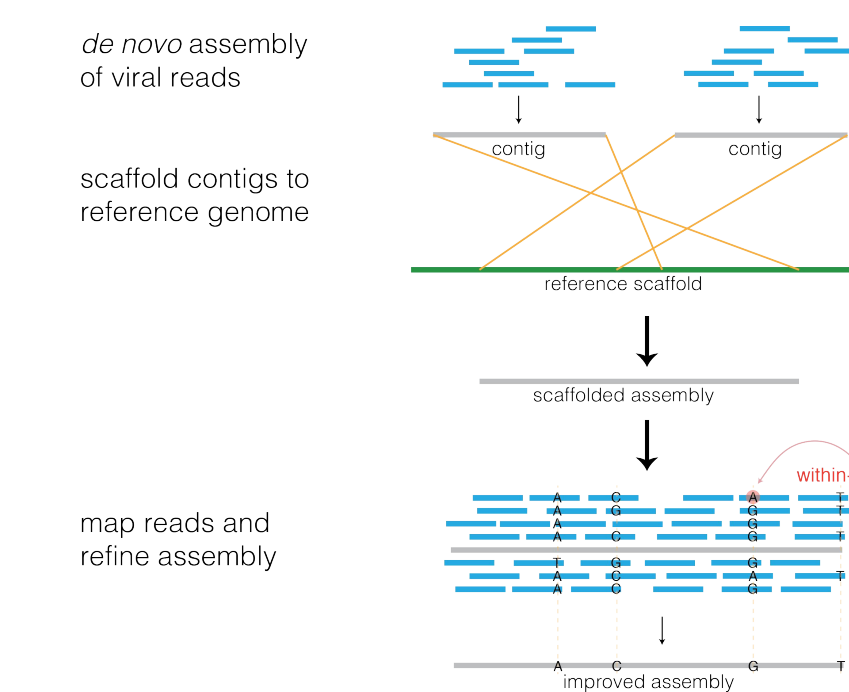
General design patterns

- Modular
- Open
- Community-driven
- Standards-based



Write pipelines in GA4GH Workflow Execution Standard (WES) compliant language

e.g. Workflow Description Language



```

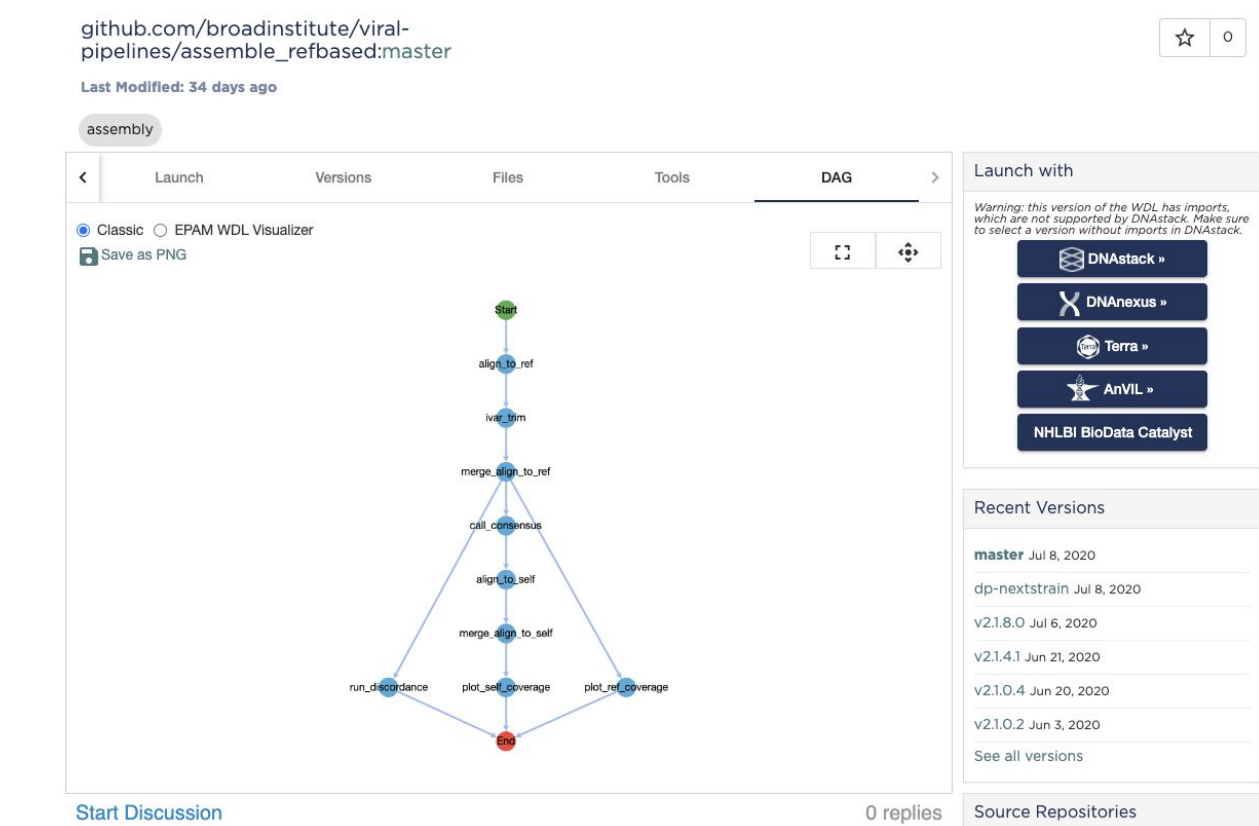
import "taxon_filter.wdl" as taxon_filter
import "assembly.wdl" as assembly

workflow assemble_de novo_with_deplete {
  call taxon_filter.deplete_taxa {
  }
  call taxon_filter.filter_to_taxon {
  }
  call assembly.assemble {
  }
  call assembly.refine_2x_and_plot {
  }
}
    
```

Distribute pipelines via GA4GH Tool Registry Service (TRS) compliant service

e.g. [dockstore.org](https://dockstore.org)

- 1-click launch on many cloud platforms
- Or run locally
- Collections can be curated by independent groups (e.g. PHA4GE)



And achieve true pipeline portability

GA4GH compute APIs: federation that works today

