



# Federated Data Access

In Four Acts.

<https://www.slideshare.net/benbusby/containerized-attribute-indexing-and-graph-g-enomes-for-federated-data-access>



# Scenes

Act 1: Virus Characterization and Discovery (x2!)

Act 2: Genome Graphs

Act 3: Annotation of Haplotypes (and Graphs)

Act 4: Indexing Data for Federated Discovery on any  
Platform, Anywhere in the World

Act 5: Epilogue: Metadata Matters



# Disclosures

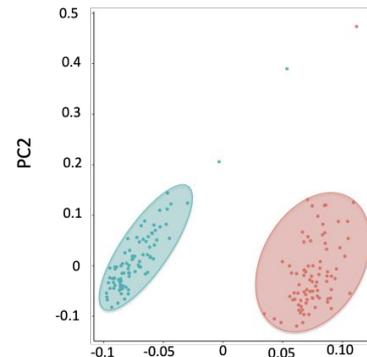
The idea and material presented here is not necessarily the view of NCBI, NLM, NIH or any other federal agency. Some of the underlying work was supported by the Intramural Research Program of the NLM and various other federal agencies.

Ben Busby is contracted to NCBI through Medical Science and Computing (MSC).

Ben Busby is also an advisor to:

- Johns Hopkins
- Ariel Precision Medicine
- Deloitte

through Mountain Genomics, a company headquartered in Pittsburgh, PA.



# Overview

---

As the volume of publicly available genomic data expands, it is becoming increasingly clear the SRA and other public repositories play an evermore important role of being stewards, allowing researchers to leverage the statistical and discovery power represented in huge arrays of datasets. That said, these repositories must not simply become 'bags of data', but indexed repositories where data can be found, approximately assessed for quality, mixed with other data, and most importantly, used by researchers to ask fundamental biological and biomedical questions.

Nearly as important, we must provision reproducible workflows, such that investigators can go from finding data to answering questions as simply and quickly as possible. These workflows will range in complexity from basic blast searching, variant calling and annotation, and transcript counting to building genome graphs and discovering novel viruses and back-spliced RNA. While doing all this we must be cognisant that while building a simple GUI interface for these types of analysis is impractical and likely impossible, the onus is on us to teach fledgeling biological data scientists to apply computational tools to their existing biological questions.



# Federation of Sequencing Data in the Cloud:

---

## *Four Illustrative Topic Areas*

1

Virus Characterization and Discovery.

*Generation of a cloud-based indexing system to allow investigators to identify data sets of interest based on taxonomic, gene and protein domain profiles*

2

Genome Graphs

*Generation of simple, usable systems to not compare an individual patient or organism to another -- or small group of -- individual, but to an entire community, dramatically compress data, and immediately find “toxic paths”.*

3

Annotation of Haplotypes (and Graphs)

*Annotation of haplotypes, instead of graphs, to allowing investigators to query complex disease*

4

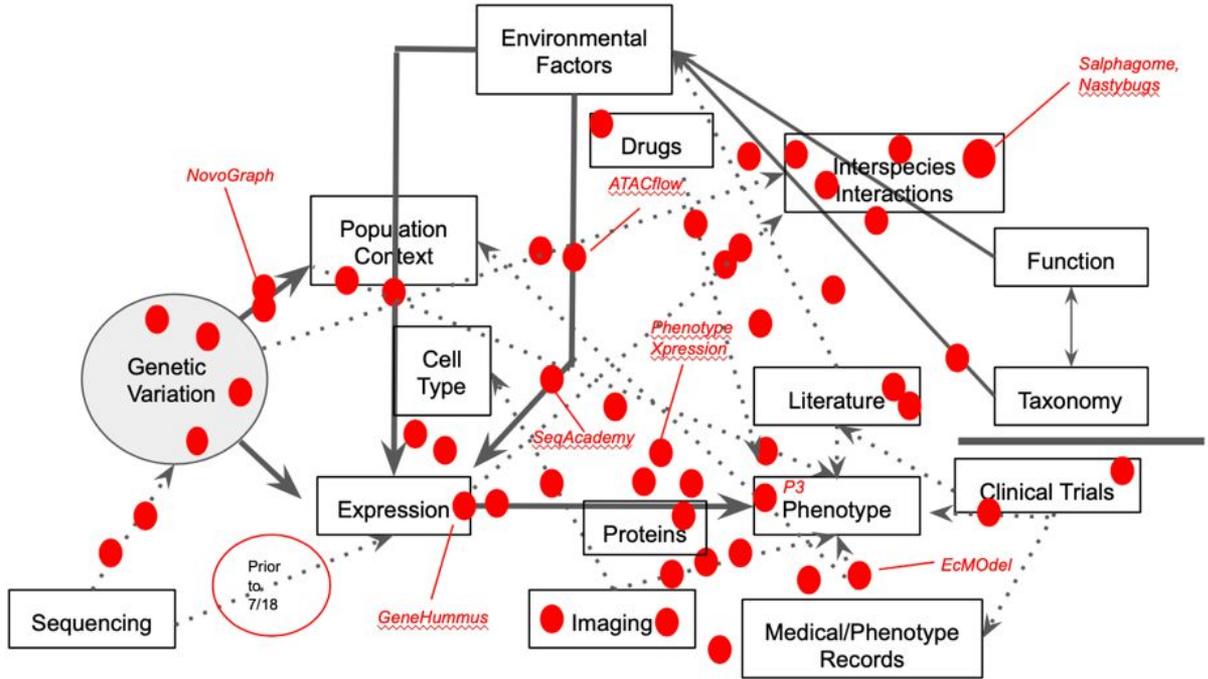
Indexing Data for Federated Discovery on any Platform, Anywhere in the World

*Flexible presentation (API) of viral protein domains, host-pathogen interactions and eventually graph loops as proof-of-principle data federation.*

### Previous Hackathons (commits 7/1/2018 - 1/1/19 only)

Human, Plant and Animal Genomes

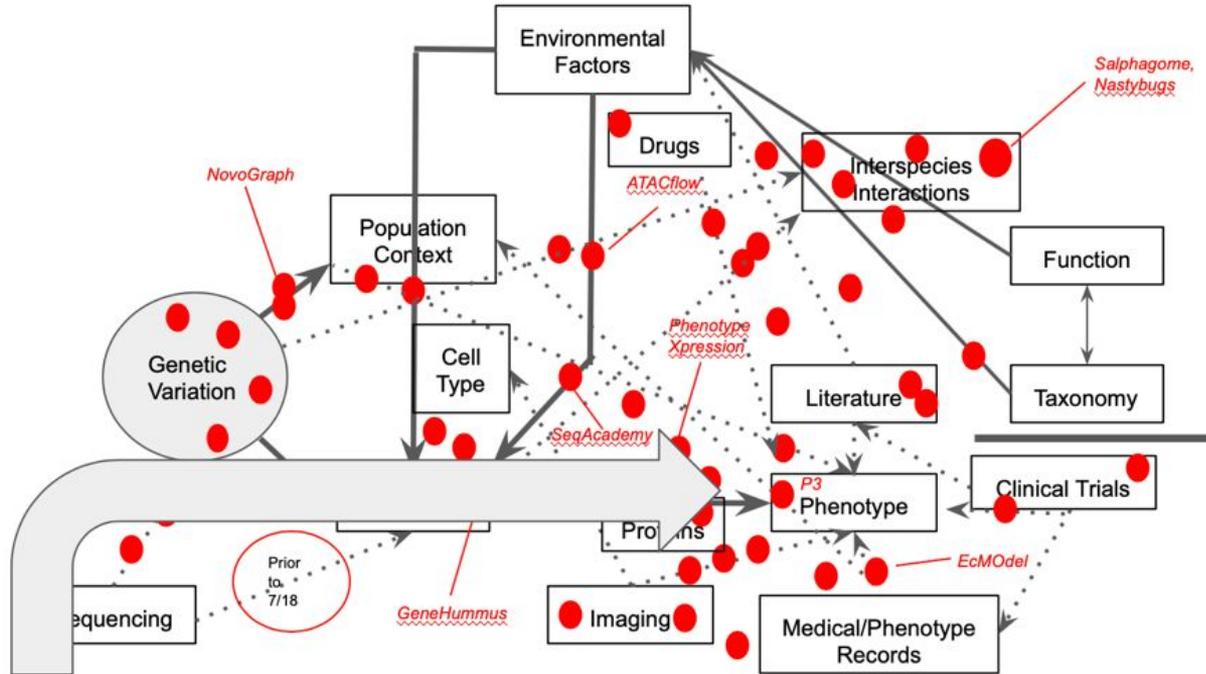
Bacterial, Viral, Fungus and Other Genomes

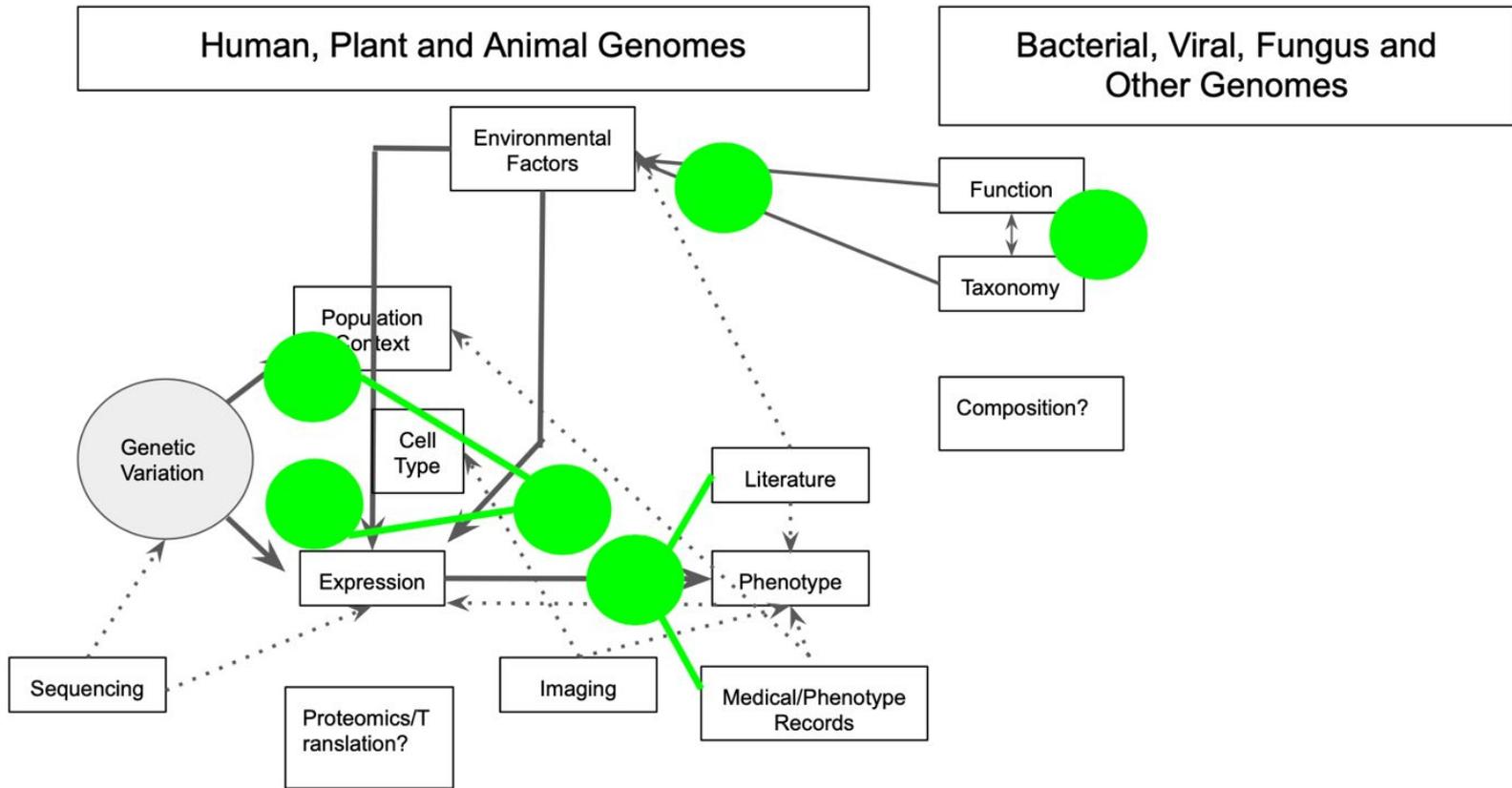


## Previous Hackathons (commits 7/1/2018 - 1/1/19 only)

Human, Plant and Animal Genomes

Bacterial, Viral, Fungus and Other Genomes





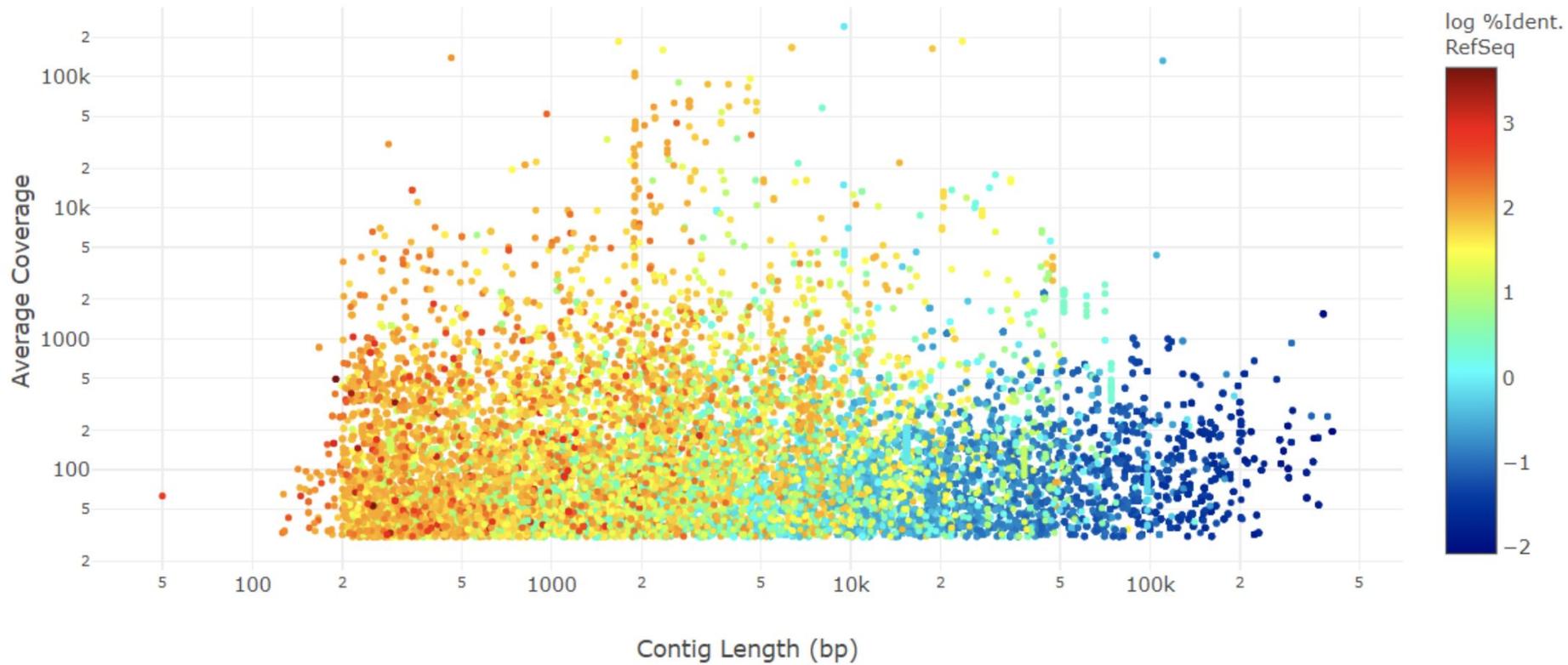


# Virus Characterization and Discovery

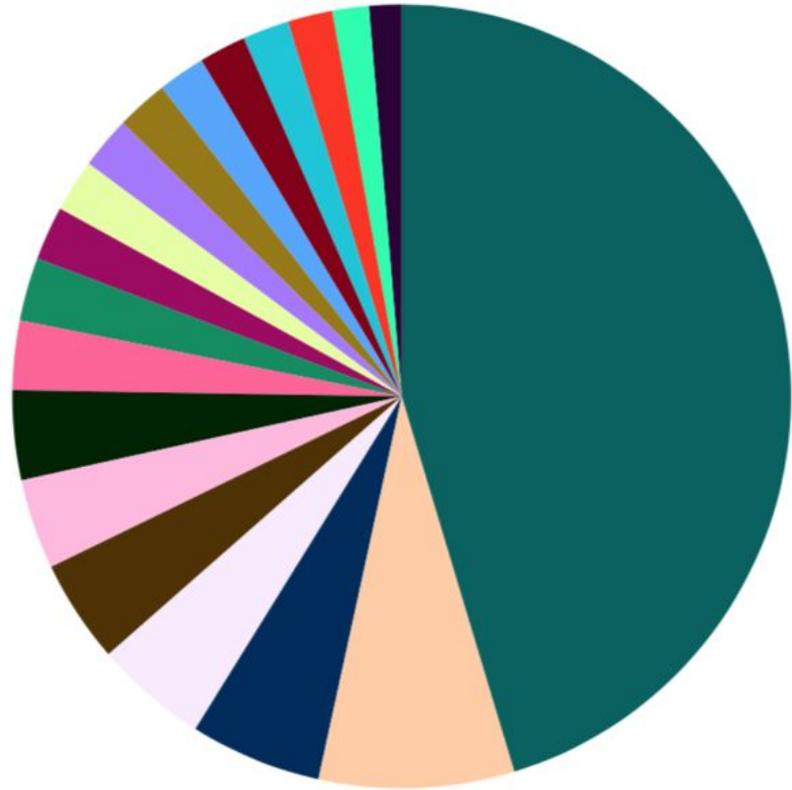
Powered by:

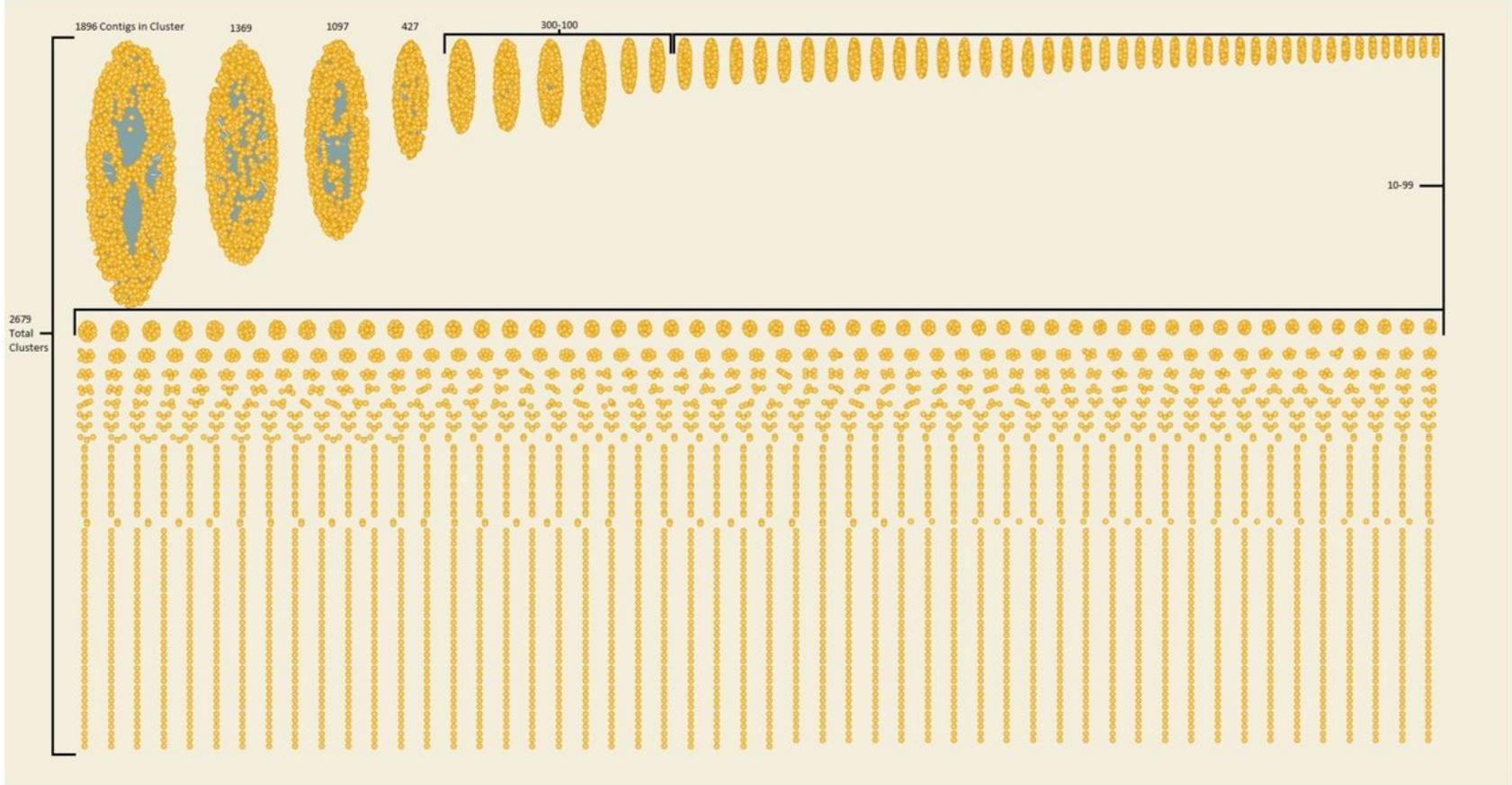


We have extracted known viruses, heretofore unknown family members, sequences identifiable as viral by the protein domains that decorate their proteins, and the true genomic “dark matter” and built a prototype index for viral signatures across the vast metadata space within SRA (~90,000-150,000 datasets). This is an example of how a researcher can reach across a massive data repository with a very simple bioinformatic tool and extract dramatic results, except in this case, we have pre-built this index for researchers working on almost any kind of virus, including noroviruses, hemorrhagic fever-causing viruses, and bacteriophages, the most abundant organisms on earth.



- uncultured crAssphage - 3398 contigs
- Enterobacteria phage HK630 - 608 contigs
- Enterobacteria phage P88 - 409 contigs
- Stx2-converting phage 1717 - 351 contigs
- Enterobacteria phage mEp460 - 320 contigs
- Enterobacteria phage cdtI - 279 contigs
- Enterobacteria phage phiP27 - 277 contigs
- Enterobacteria phage SfV - 214 contigs
- Escherichia virus P1 - 196 contigs
- Shigella phage SfIV - 167 contigs
- Escherichia phage APCEc01 - 163 contigs
- Escherichia phage 121Q - 162 contigs
- Escherichia phage PBECO 4 - 161 contigs
- Salmonella phage 118970\_sal3 - 147 contigs
- Enterobacteria phage fiAA91-ss - 146 contigs
- Escherichia phage TL-2011b - 145 contigs
- Enterobacteria phage YYZ-2008 - 139 contigs
- Enterobacteria phage BP-4795 - 114 contigs
- Escherichia phage D108 - 101 contigs

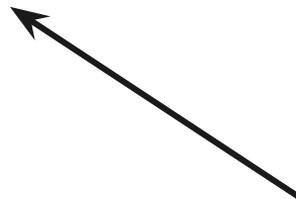
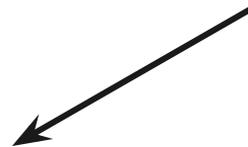






# Genome Graphs

Initially leveraging standard workflows for annotation of primary datasets in cloud infrastructure mapped to linear genomes, we will likely transition to genome graphs over the next ten years. This will allow us to compress reads, do phasing automatically, do reference-guided assembly on complex genomes, and detect “toxic paths” automatically. Most importantly, this will allow us to see complex genotype - phenotype interactions easily.



# Graph Genomes!!!

START HERE

Step 0: Get BAM alignments for each assembly



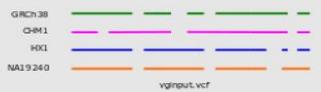
Step 1.1: Do pairwise alignment against reference



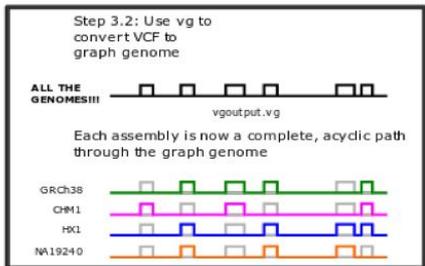
Step 1.2: Join pairwise alignments into single FASTA file



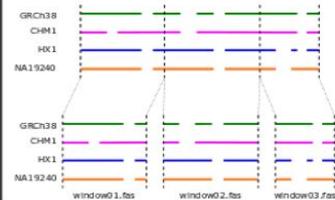
Step 3.1: Convert BAM file to VCF (in parallel)



Step 3.2: Use vg to convert VCF to graph genome



Step 2.1: Identify windows and extract one fasta file for each window



↑ parallelized steps

Step 2.3: Concatenate all alignments into single BAM file



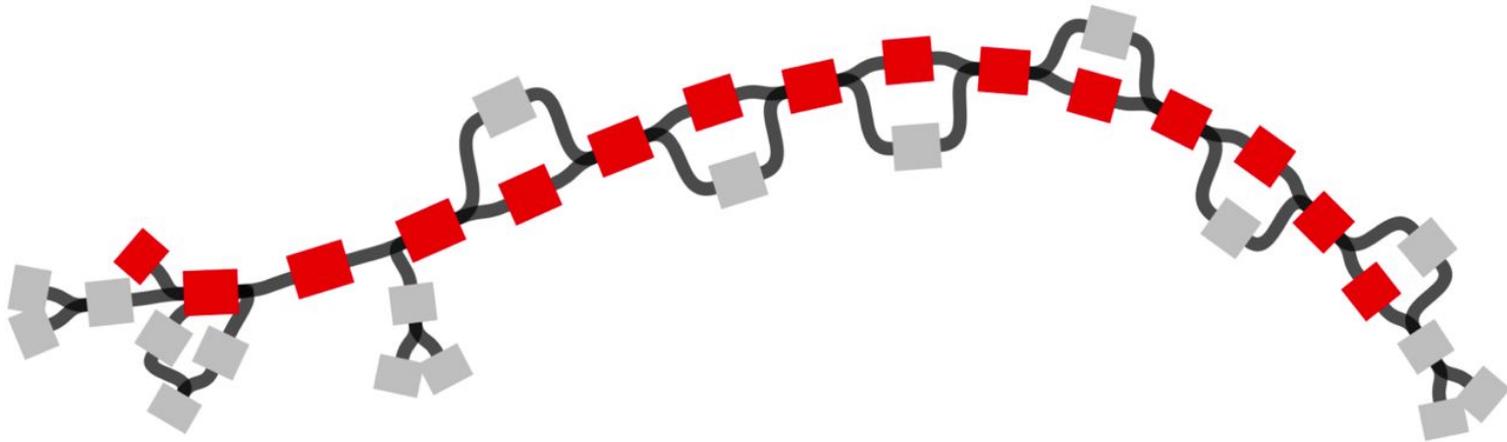
Step 2.2.2: Convert all alignment FASTA output files to BAM format



Step 2.2.1: Perform multiple sequence alignment for each window



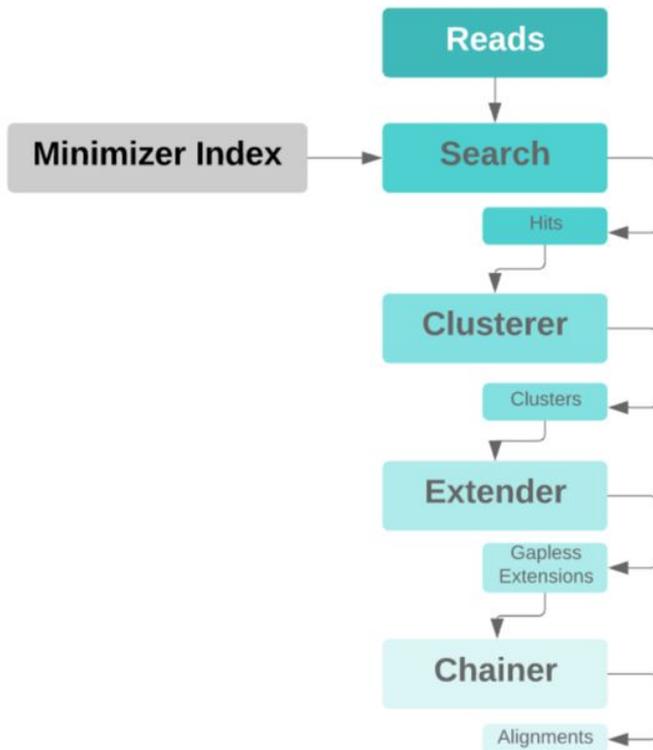
Evan Biederstedt, Alex Dilthey  
many others



The path of GRCh38 through a graph! For details check the DS folder!

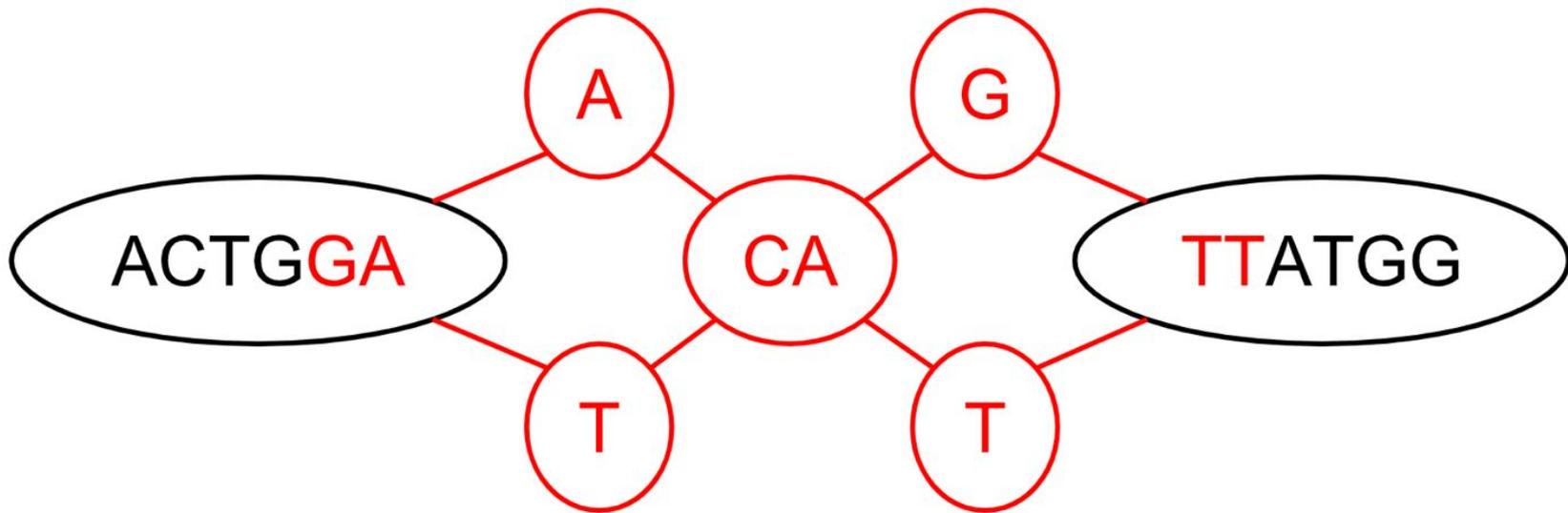
<https://github.com/NCBI-Hackathons/TheHumanPangenome/tree/master/DS>

Erik Garrison, many others



A Faster, Better Short-Read Mapper with Hit Chaining. Now with more corn! See the Giraffe folder for details!

<https://github.com/NCBI-Hackathons/TheHumanPangenome/tree/master/Giraffe>

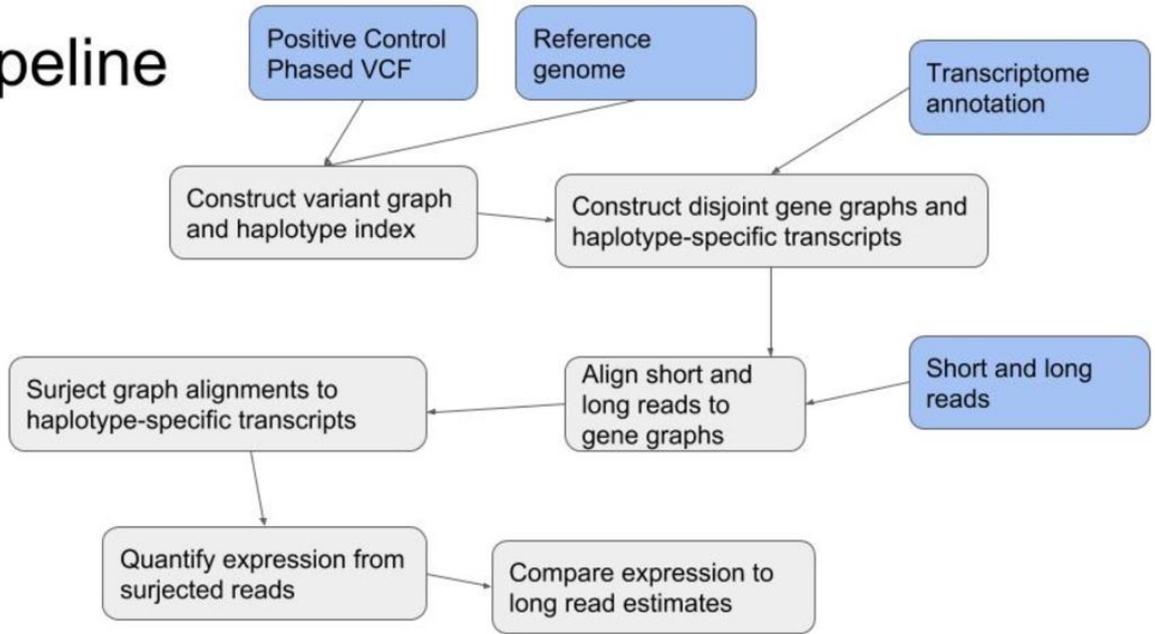


Most Genome Annotations can be Imported from gff3 (to ggff!) quite easily! See the annotation folder for details!

<https://github.com/NCBI-Hackathons/TheHumanPangenome/tree/master/annotation>

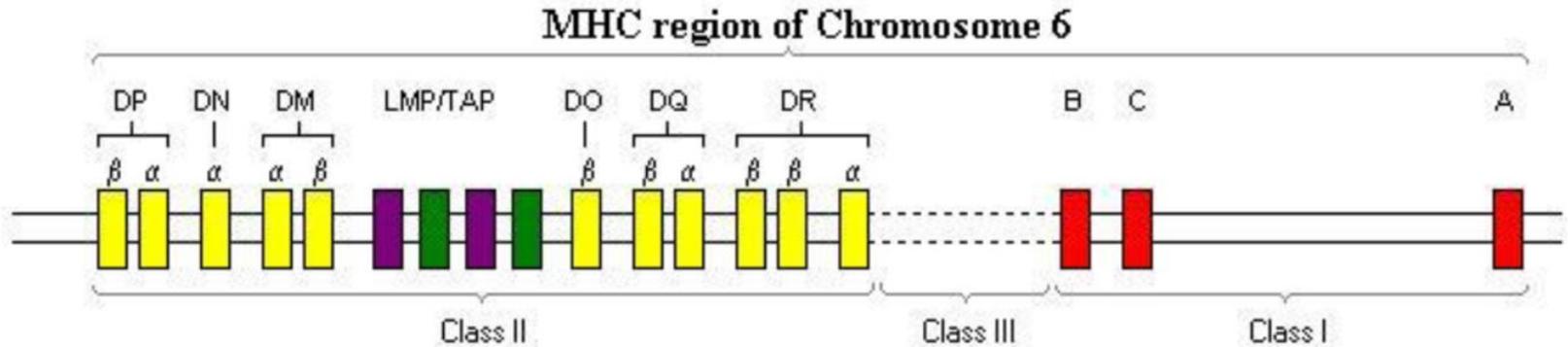


# Pipeline



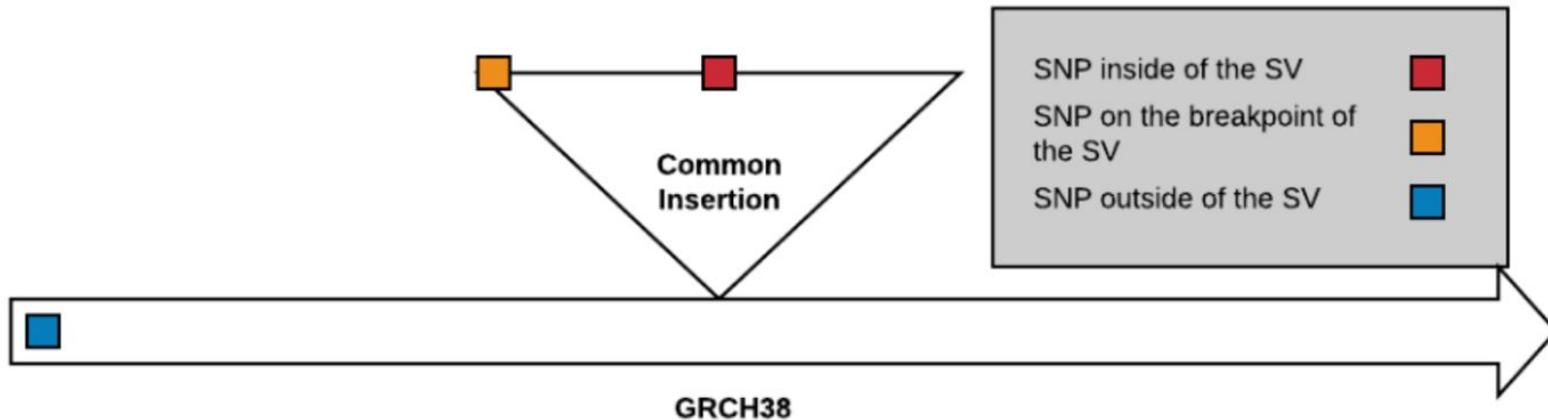
Allele-Specific Expression can be calculated in a lightweight fashion! Grab the workflow in the RNA folder!

<https://github.com/NCBI-Hackathons/TheHumanPangenome/tree/master/RNA>



We Mapped the MHC Region of HG002 to a Graph with Long Reads, Long Reads, 10X and love. Check out the MHC folder!

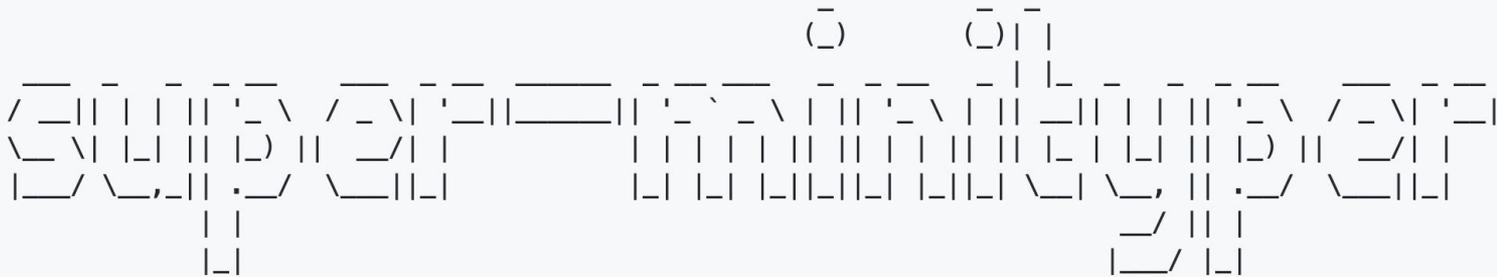
<https://github.com/NCBI-Hackathons/TheHumanPangenome/tree/master/MHC>



Calculate Mutations In, Outside of, and at Breakpoints of Structural Variants! See the SV folder for Details!

<https://github.com/NCBI-Hackathons/TheHumanPangenome/tree/master/SV/HG002>

Fritz Sedlazeck, many others

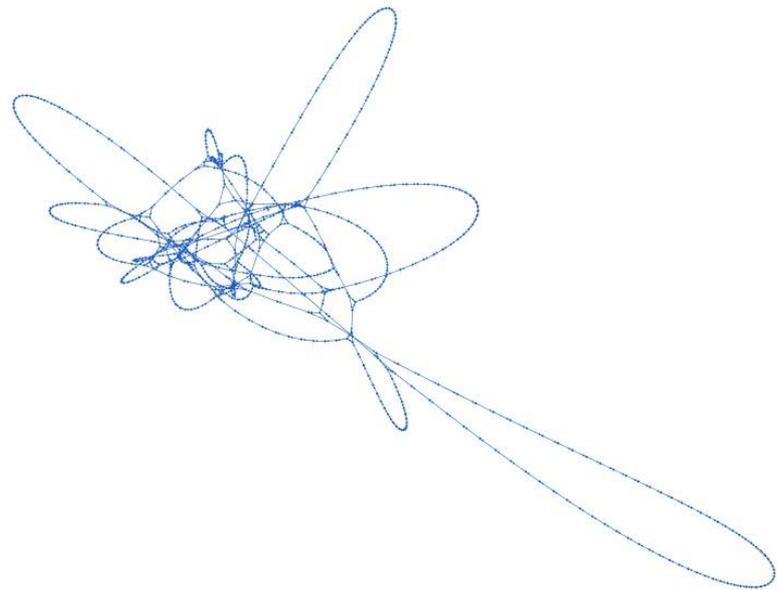
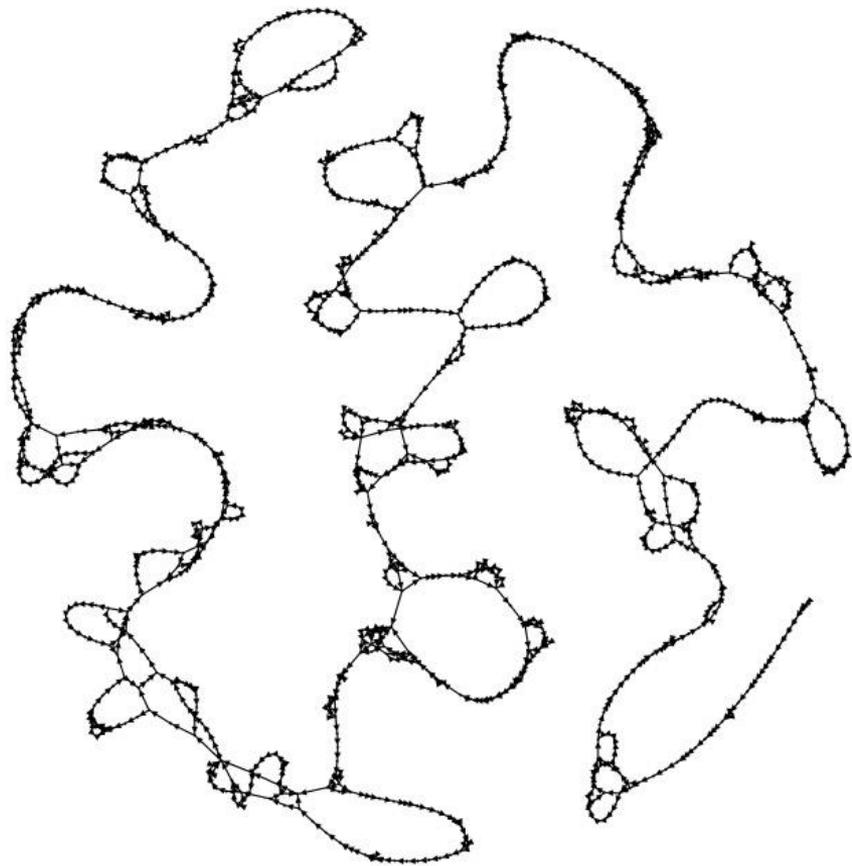


`super-minityper` is a set of cloud-based workflows for constructing SV graphs and mapping reads to them.

Structural variants frustrate read mapping because aligners often choose not to map read portions which map very distantly. Graphs allow incorporating known variants, including large ones, and then mapping directly to these. While this has been shown to reduce reference bias and improve read mappings when a sample contains variants in the graph, constructing graph genomes and operating on them has historically been difficult and time-consuming.

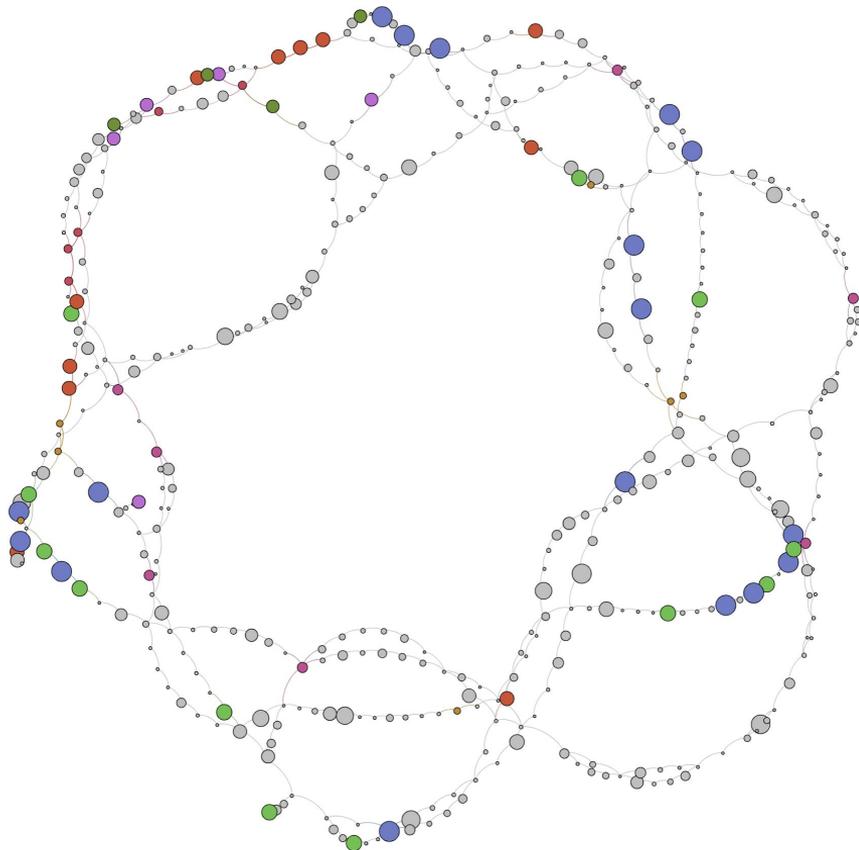
We present a set of cloud-based workflows — composed mostly of preexisting and optimized tools — to construct graphs containing structural variants and map reads to them. These workflows allow users to take arbitrary SV calls, construct a graph, and map reads to these graphs. This workflow prioritizes ease-of-use and speed, ingesting common input formats and returning results in minutes on commodity cloud VMs.

Eric Dawson, Fernanda Forterre, many others



<https://github.com/NCBI-Codeathons/SWIGG>

Jason Chin, Alex Gener, many others



## swigg\_shiny

Shiny app for visualizing genome graphs from SWIGG

### In development, current priorities:

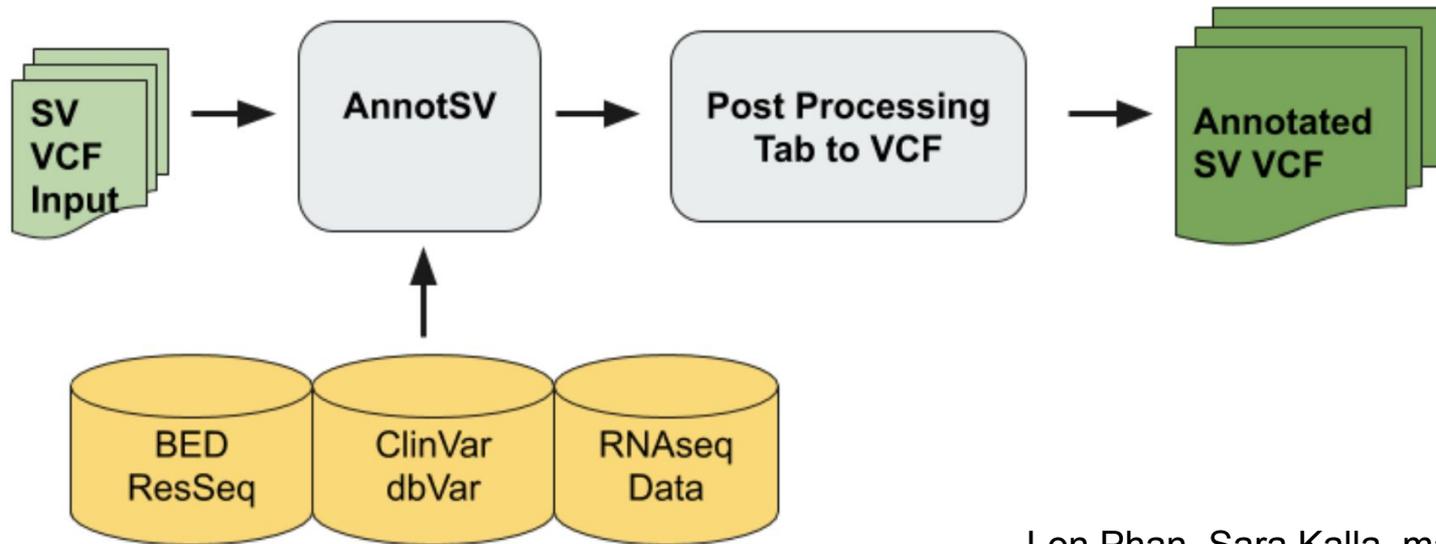
#### 1. Interactive plot

- Goal: View attributes of a node or edge when you hover over it in the plot
- Plan: Since plotly currently doesn't work with ggraph, try converting code from ggraph to igraph

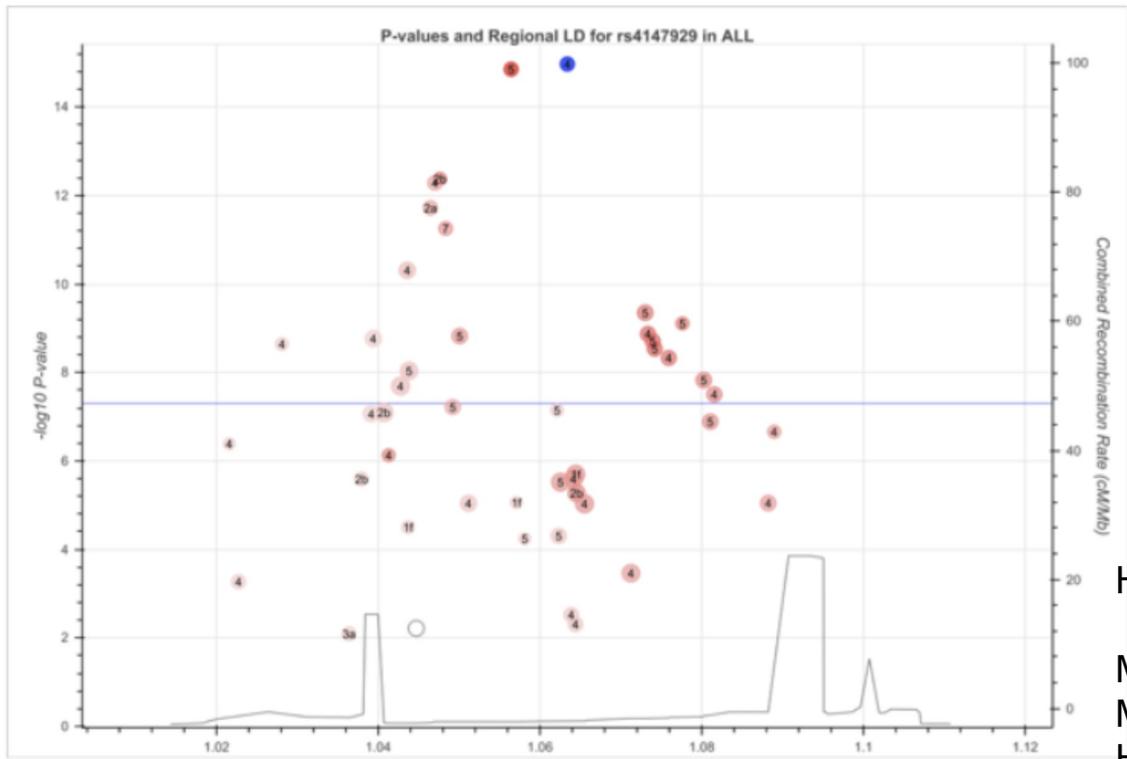
[github.com/ncbi-codeathons/virus\\_graphs](https://github.com/ncbi-codeathons/virus_graphs)

Mike Tisza, Alexis Norris, many others

# Medical Annotations of Graphs and Haplotypes

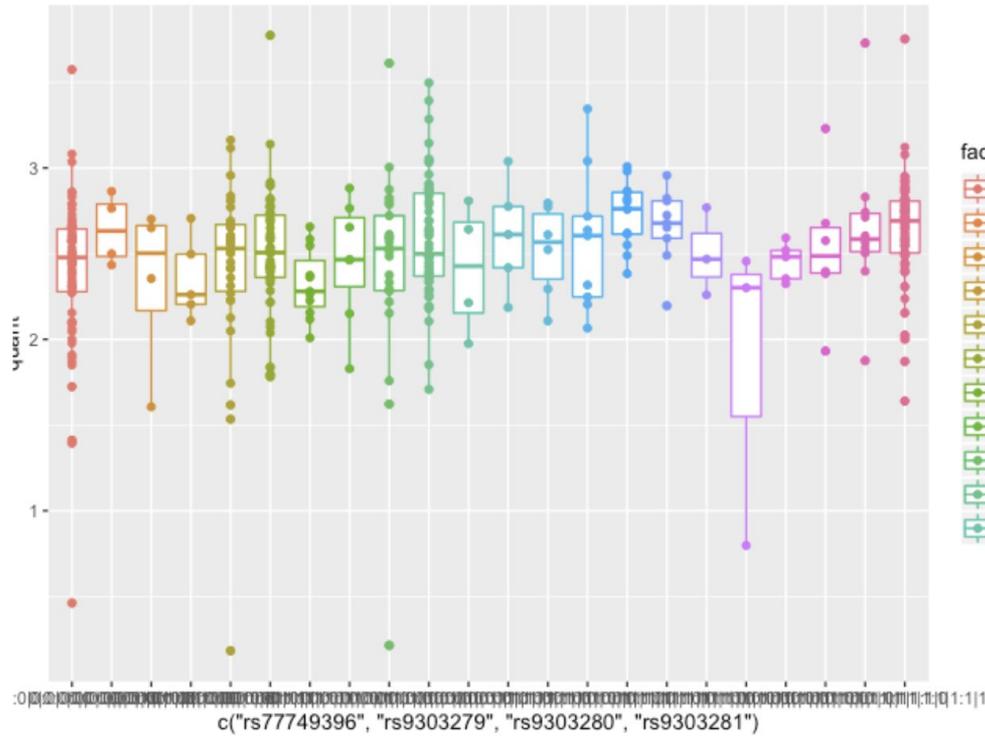


Lon Phan, Sara Kalla, many others



Haplocravat2  
Mike Smallegan, Kyle Moad, Matt Hynes-Grace, Dina Mikdadi, many others

### GSDMB expression

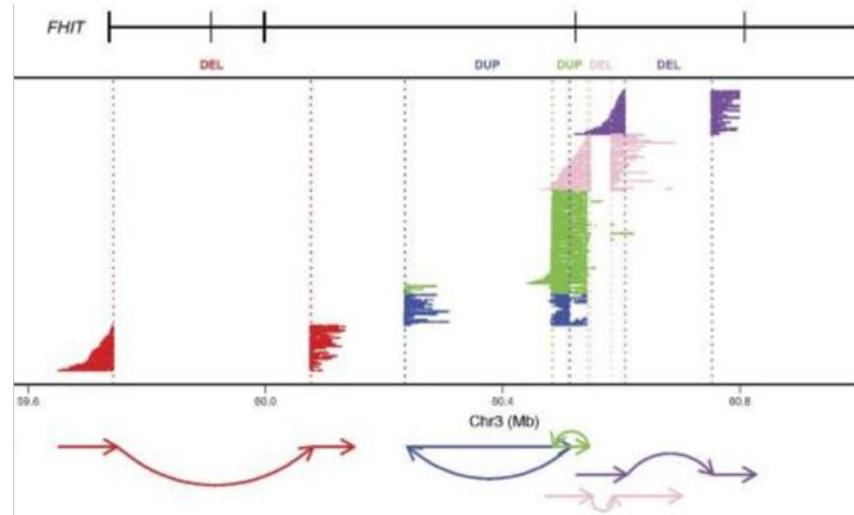
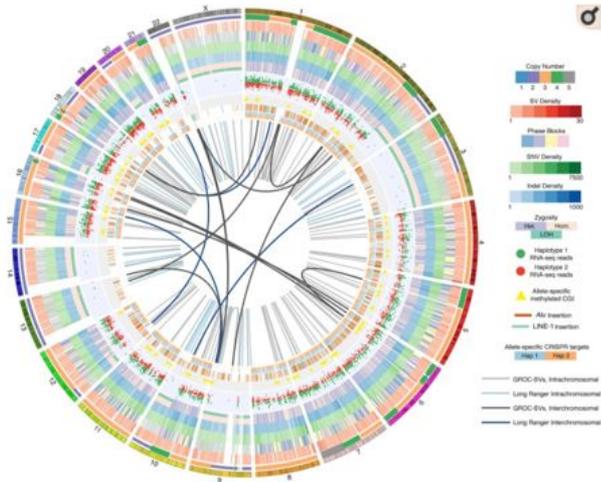


### factor(hap)

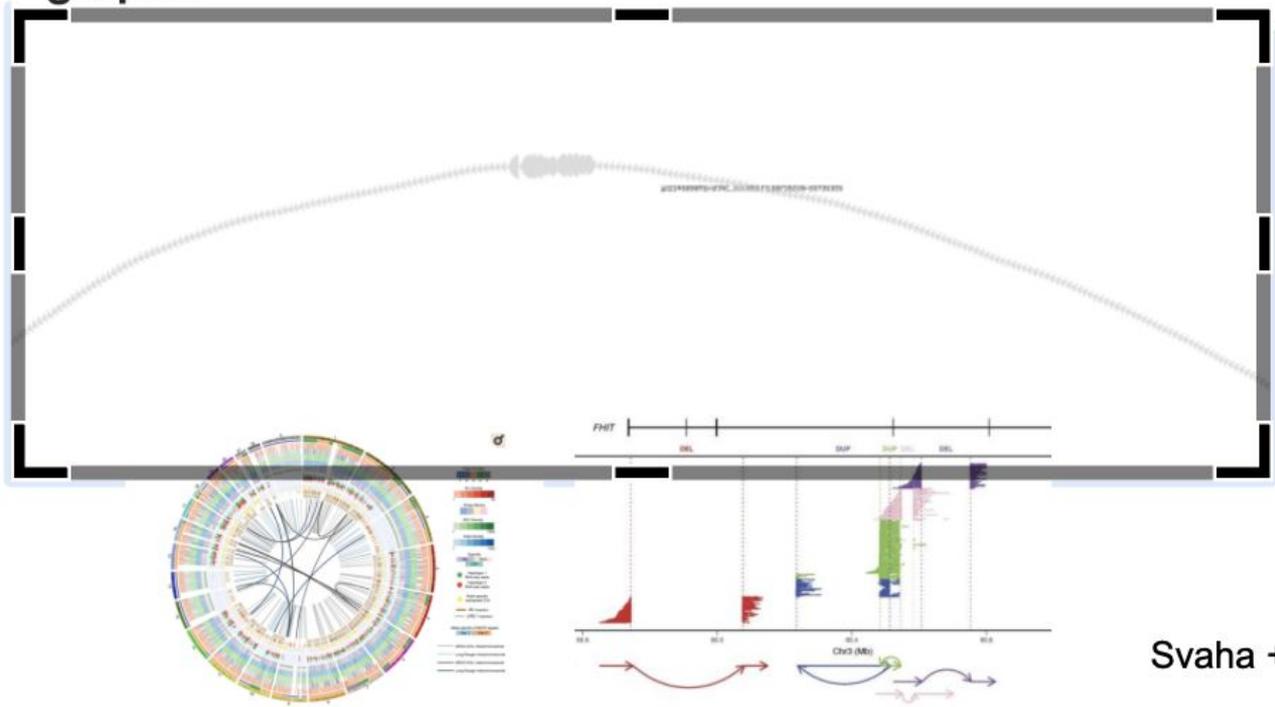
- 0|0:0|0:0|0:0:0|0:0:0
- 0|0:0|0:0|0:0:0|1|0:0:0
- 0|0:0|0:0|1:0|1:0|0:0:0
- 0|0:0|0:1|0:1|0:0|0:0:0
- 0|1:0|0:0|0:0:0|1:0|1:0:1
- 0|1:0|0:0|1:0|1:0|1:0:1
- 0|1:0|0:1|1:1|1:0|1:0:1
- 0|1:1|0:1|1|1:1|1:1|1:0
- 1|0:0|0:0|0:1|0:1|0:1|0
- 1|0:0|0:1|0:1|0:1|0:1|0
- 1|0:0|0:1|1:1|1:1|1:0
- 1|0:0|1|0:1|0:1|1:1|0
- 1|0:0|1|0:1|0:1|1:1|0
- 1|0:0|1|1:1|1:1|1:0:1|0
- 1|0:0|1|1:1|1:1|1:1|1:1
- 1|1:0|0:0|1|1:1|1:1|1:1
- 1|1:0|0:0|1|1:1|1:1|1:1
- 1|1:0|0:1|0:1|1:1|1:1|1
- 1|1:0|0:1|0:1|1:0|1:1|1
- 1|1:0|0:1|1:1|1:1|1:0:0:0
- 1|1:0|0:1|1:1|1:1|1:0:0:0
- 1|1:0|0:1|1:1|1:1|1:1:0:0
- 1|1:0|0:1|1:1|1:1|1:1:0:1
- 1|1:0|0:1|1:1|1:1|1:1:1:0
- 1|1:0|0:1|1:1|1:1|1:1:1:1

Vince Carey, John Didion, many others

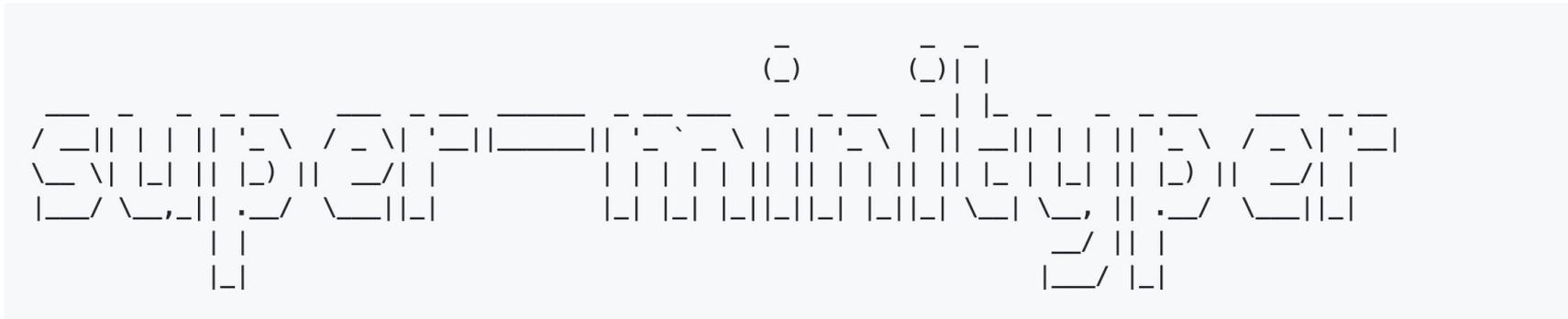
# Using p53 and K562 as examples to define toxic paths in graphs



# Using p53 and K562 as examples to define toxic paths in graphs



Svaha + Bandage



`super-minityper` is a set of cloud-based workflows for constructing SV graphs and mapping reads to them.

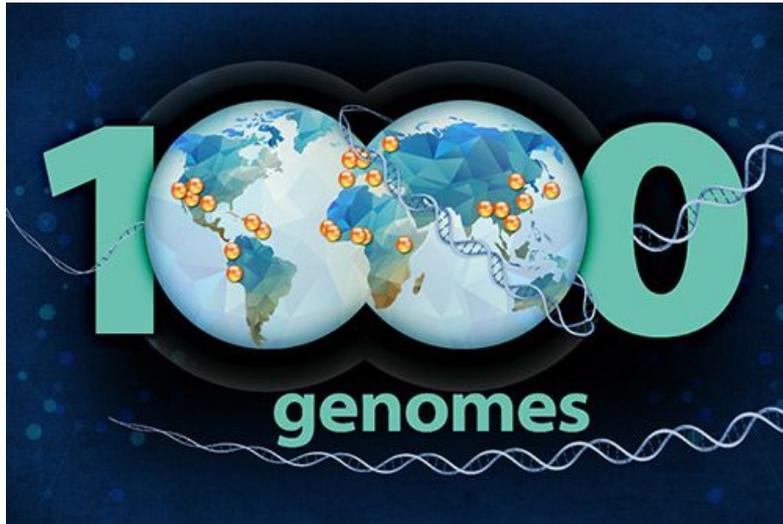
Structural variants frustrate read mapping because aligners often choose not to map read portions which map very distantly. Graphs allow incorporating known variants, including large ones, and then mapping directly to these. While this has been shown to reduce reference bias and improve read mappings when a sample contains variants in the graph, constructing graph genomes and operating on them has historically been difficult and time-consuming.

We present a set of cloud-based workflows — composed mostly of preexisting and optimized tools — to construct graphs containing structural variants and map reads to them. These workflows allow users to take arbitrary SV calls, construct a graph, and map reads to these graphs. This workflow prioritizes ease-of-use and speed, ingesting common input formats and returning results in minutes on commodity cloud VMs.

Eric Dawson, Fernanda Forterre, many others

---

# A Brief Proposal for Graph-Based Sequence Compression and Phasing (personal opinion)



+



+





# A Brief Proposal for Graph-Based Sequence Compression and Phasing (personal opinion)

=

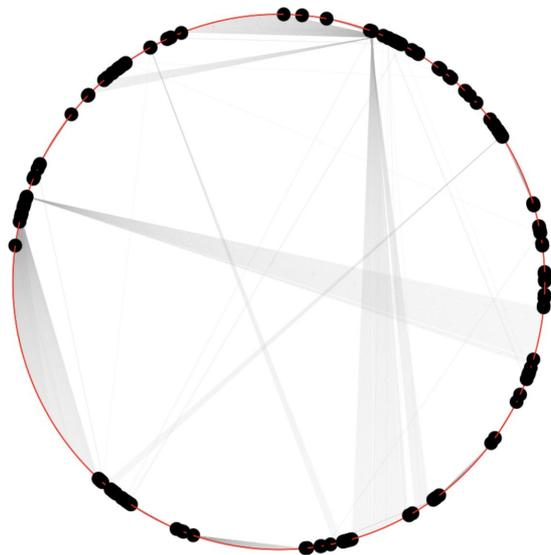


(no longer available on CD-ROM)

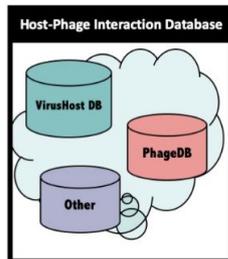


# Indexing Data for Federated Discovery on any Platform, Anywhere in the World!

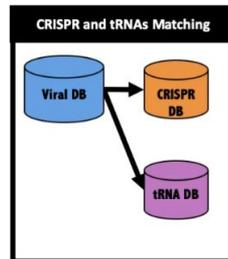
Hosts (black circles) Linked to Viruses (red circles)



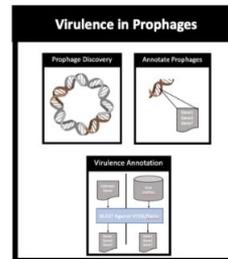
## Known Knowns



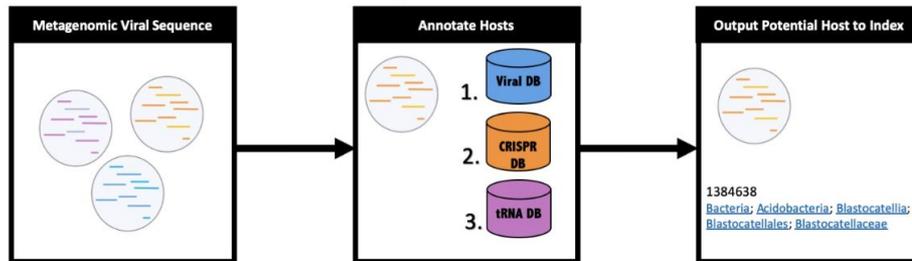
## Known Unknowns



## Genes in HPI



## User Input Workflow





# Indexing Data for Federated Discovery on any Platform, anywhere in the World!

## The\_Virus\_Index

---

A Federated Index of Virus Metadata and Hyperdata in Public Repositories

## API

---

Status: Extensible DRAFT API

build passing

<https://test.pypi.org/project/viral-index/>

### Requirements:

- `python3`
- A Google Cloud Platform (GCP) account. Please see [GCP's getting started guide](#) if you are new to GCP.

Christiam Camacho, Sej  
Modha, Alex Efremov,  
Joan Marti-Carreras

# Indexing Data for Federated Discovery on any Platform, anywhere in the World!

## The\_Virus\_Index

A Federated Index of Vir

### API

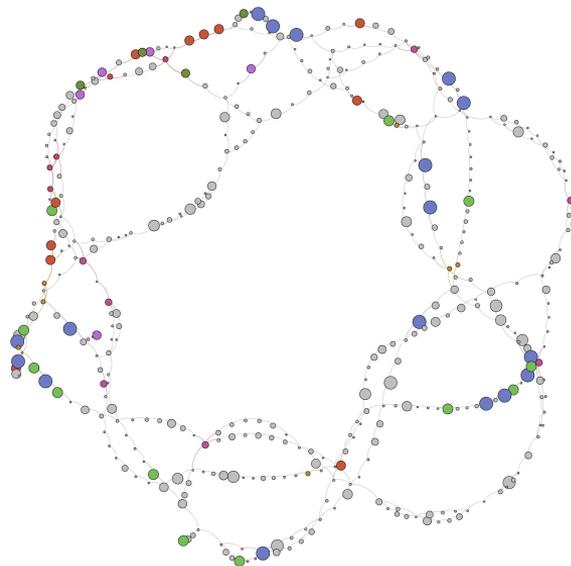
Status: Extensible DRAF

build passing

<https://test.pypi.org/proj>

#### Requirements:

- python3
- A Google Cloud Plat



Christiam Camacho, Sej  
Modha, Alex Efremov,  
Joan Marti-Carreras

[guide](#) if you are new to GCP.



# The Importance of Metadata

If biologists leverage the data indices or pipelines for the illustrative subject areas listed here, without metadata, they will simply be doing more indexing and cataloging. For maximum utility, the primary data must be accompanied by rich metadata that puts the samples in question in context, not just in terms of their technical origins, but also their biological origins; for example, their disease state, tissue type, and the precise geographic location, not of the sequencing, but where the individual organism lives.

## SpOdgi

---

pypi package **0.0.2**

Use a general graph query language SPARQL to investigate genome variation graphs!



<https://ncbi-codeathons.github.io>

<https://biohackathons.github.io>

(not an NCBI site)

<https://www.github.com/ncbi-hackathons>

(Archive)

<https://www.github.com/ncbi-codeathons>

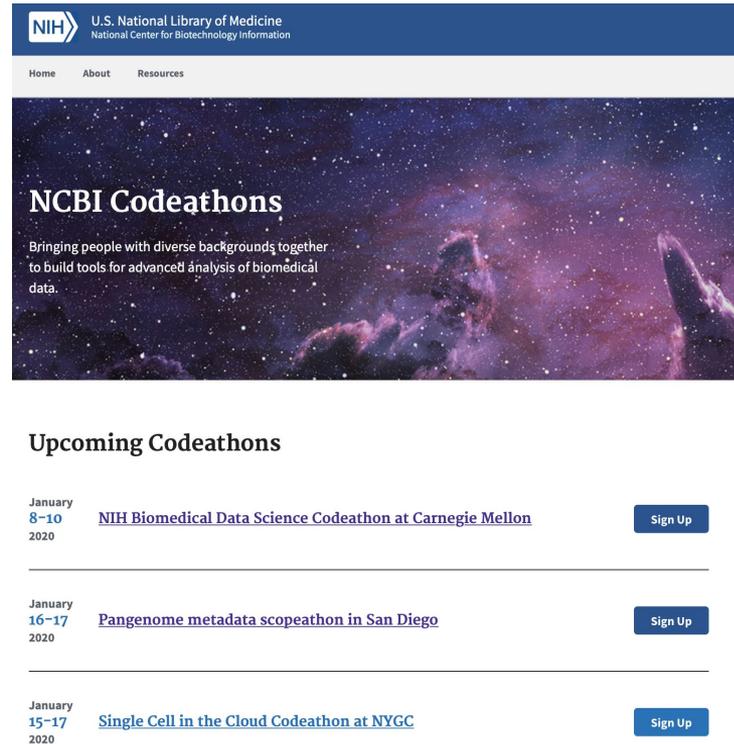


<https://ncbi-codeathons.github.io>

<https://biohackathons.github.io>  
(not an NCBI site)

<https://www.github.com/ncbi-hackathons>  
(Archive)

<https://www.github.com/ncbi-codeathons>



The screenshot shows the NCBI Codeathons website. At the top is the NIH logo and the text "U.S. National Library of Medicine National Center for Biotechnology Information". Below this is a navigation bar with "Home", "About", and "Resources". The main header features a space-themed background with the title "NCBI Codeathons" and the tagline "Bringing people with diverse backgrounds together to build tools for advanced analysis of biomedical data." Below the header is a section titled "Upcoming Codeathons" which lists three events:

- January 8-10 2020: NIH Biomedical Data Science Codeathon at Carnegie Mellon (Sign Up)
- January 16-17 2020: Pangenome metadata scopeathon in San Diego (Sign Up)
- January 15-17 2020: Single Cell in the Cloud Codeathon at NYGC (Sign Up)