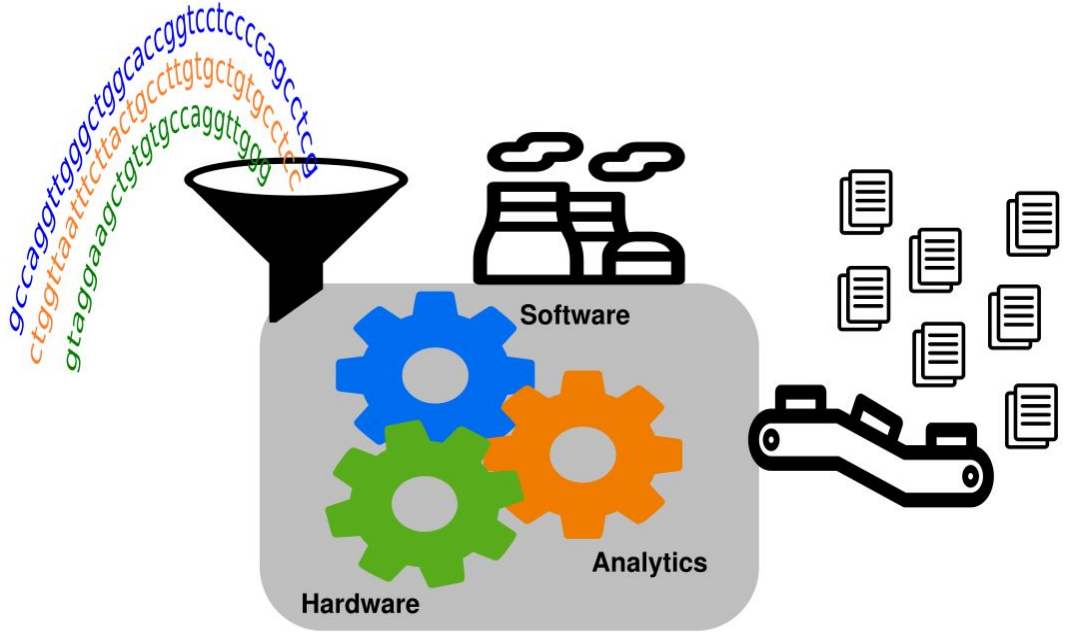
Enabling Economical Genome Analyses through Optimization and Scalable Workflows

ParslFest 2020

Akila Ravihansa Perera
Research Assistant (Software Developer) - Pitt Lab
Cancer Science Institute of Singapore
National University of Singapore

The genomic data science machine

Scaling cancer research



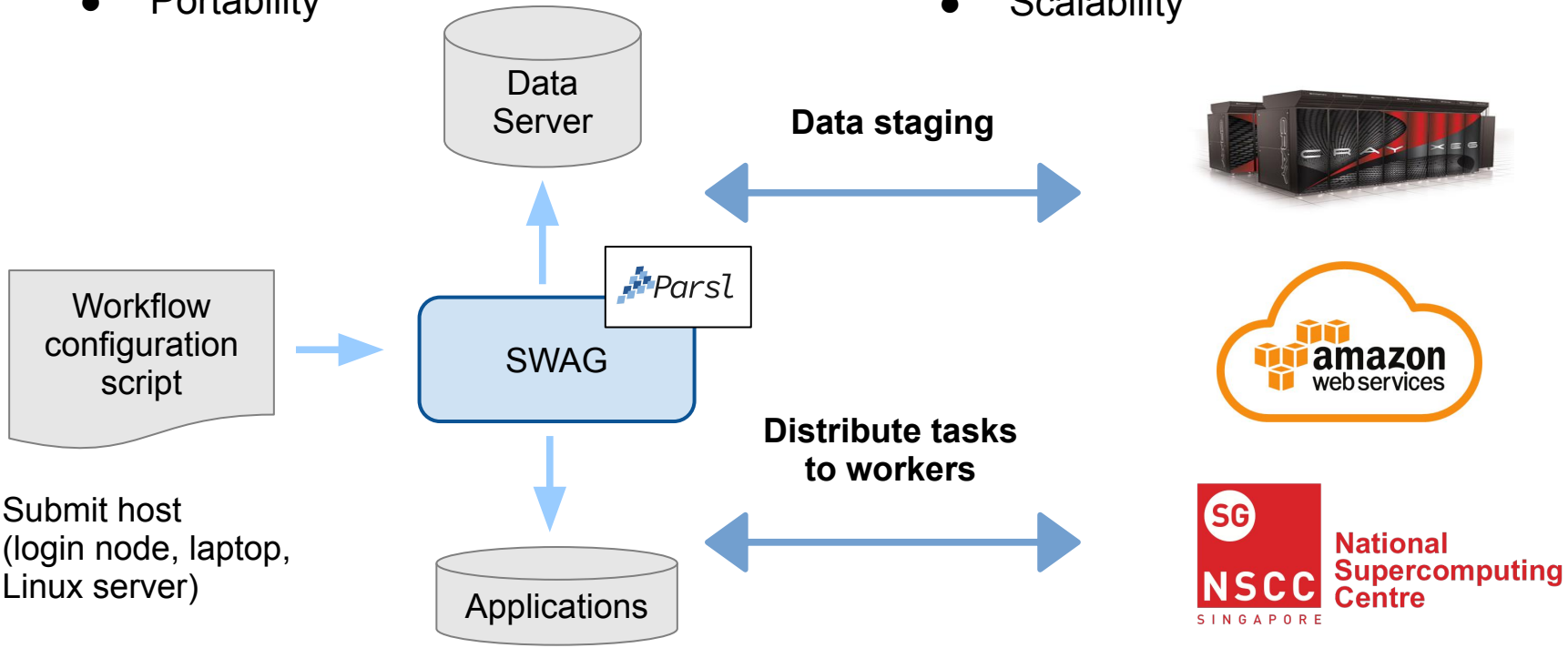
Cloud Platforms

Supercomputing Clusters

Compute Cost

SWAG: Scalable Workflows for Analyzing Genomes

- Transparent parallelism
- Portability
- Robust to failures
- Scalability



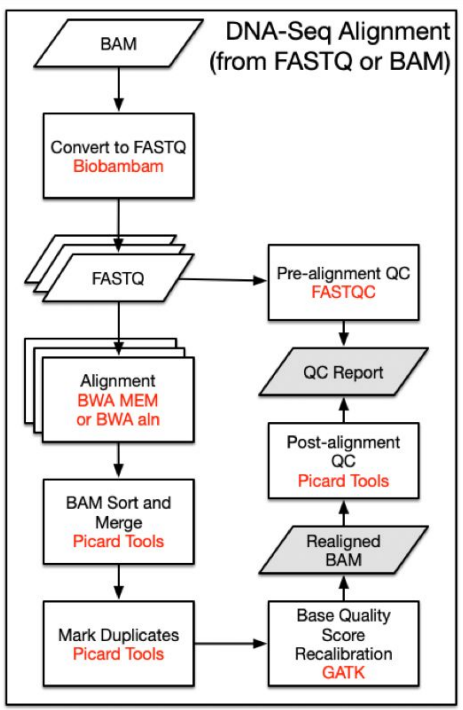
Integrating local and public DNAseq samples

Problem: we want to compare our collection of locally generated DNAseq samples to those in public repositories

Possible solutions:

1. Download all raw data and process locally
    a. Computationally and monetarily expensive
2. Take data as is (e.g. different pipelines on the respective datasets)
    a. Highly prone to technical artifacts; datasets are not comparable
3. Emulate a trusted pipeline locally

Harmonizing to the Genomic Data Commons

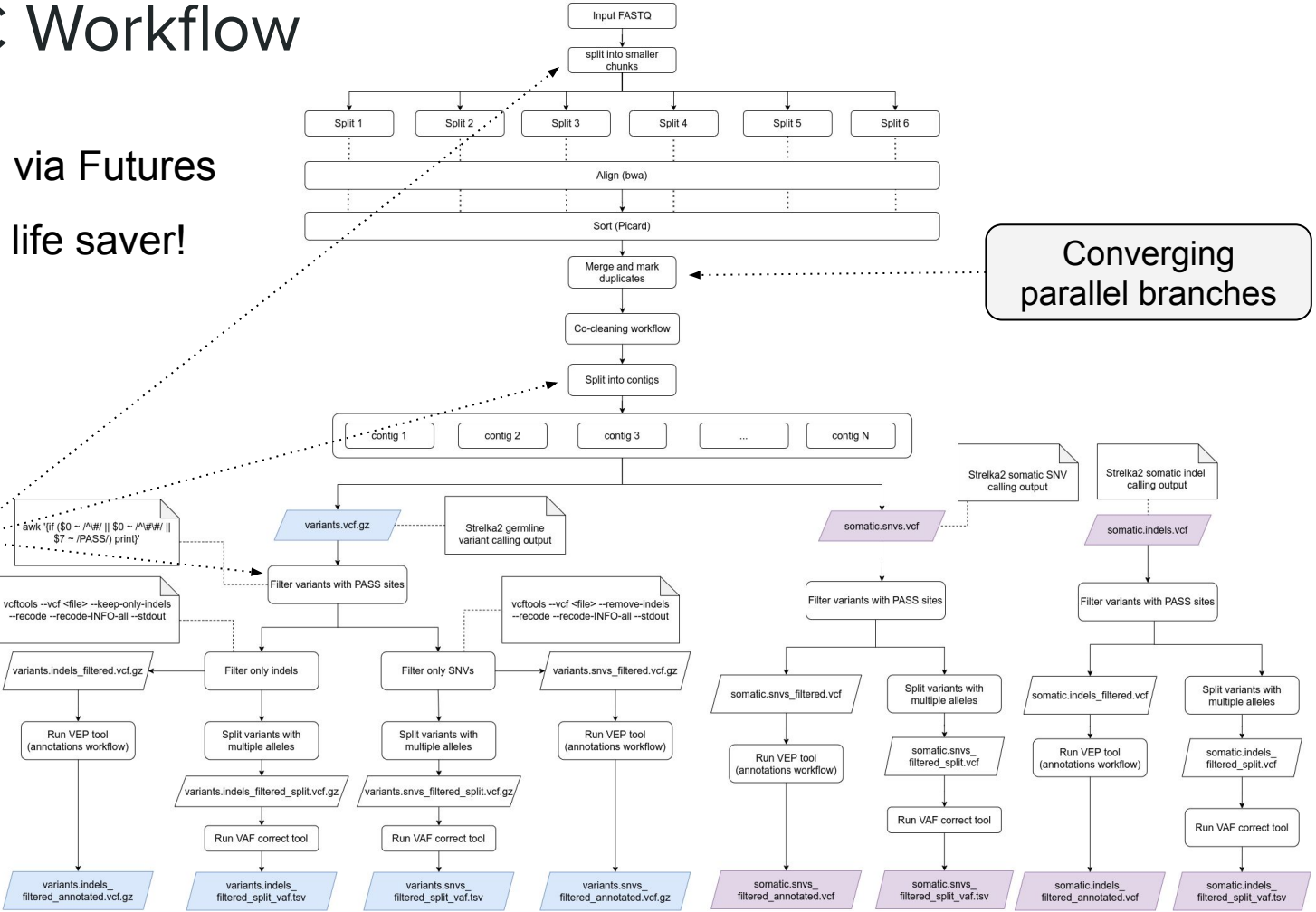


*** hg38 used as ref genome**

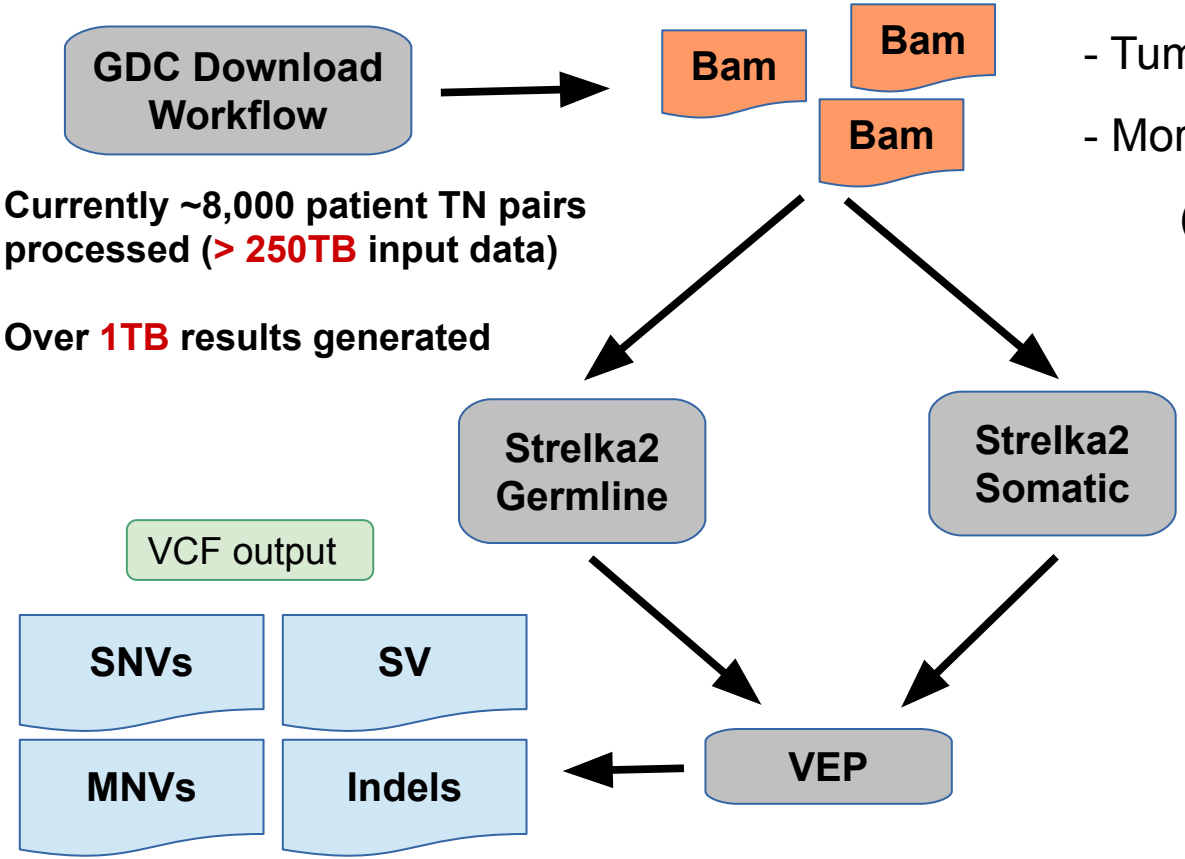
- Tens of thousands of sequenced (panel, exome, & genome) cancer samples
- All samples processed through the same pipeline
- A scalable GDC analytical workflow not yet provided

SWAG - GDC Workflow

- Parsl app chaining via Futures
- Checkpointing is a life saver!



SWAG - Supplementing Data from GDC



Currently ~8,000 patient TN pairs processed (> 250TB input data)

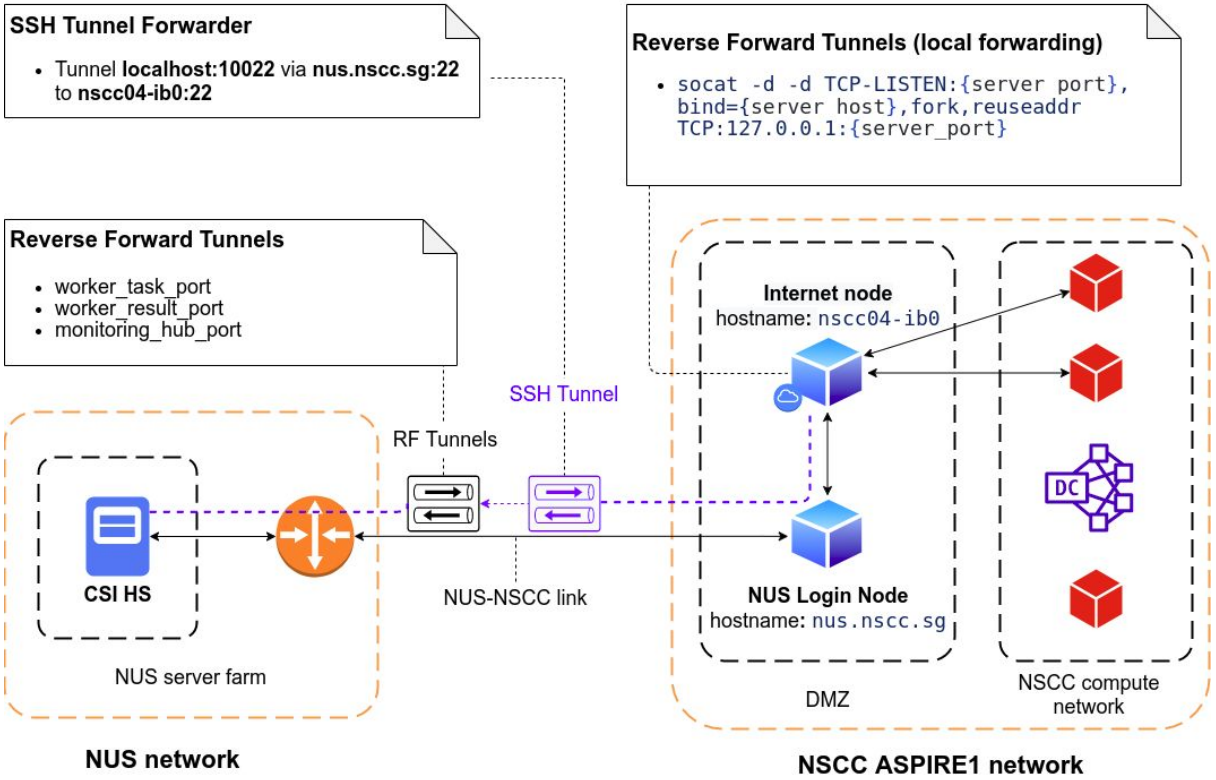
Over 1TB results generated

- Tumor-normal (TN) pairs from TCGA
- More than **10k** cancer patients

  (**33** cancer types)

ASIPRE1
(~700 nodes & ~30,000 cores)

GDC Download Workflow - Infrastructure



- Use LocalProvider with SSHChannel to execute apps on remote nodes
- Create SSH reverse forward tunnels to connect workers to interchange
- Useful when Parsl cannot be executed on a remote instance directly
- Workers should be able to resume work without significant wastage

Wishlist

- More flexibility in packing tasks across multiple executors
  - Deal with task scheduling queue limitations
  - ASPIRE1 medium queue - 24hrs timeout, N1 cpus max
  - ASPIRE1 long queue - 120hrs timeout, N2 cpus max , ($N1 \neq N2$)
- Improved support for remote task execution
  - Non-shared file system between Parsl and workers

Acknowledgements

Pitt's Lab, CSI

Dr. Jason Pitt
PI

Vinay Warrier
Software Engineer
(Data Analytics)

Stefanus Lie
PhD Student

Hannan Wong
PhD Student

External Collaborators

Anna Woodard, PhD
Postdoctoral Scholar,
University of Chicago

Ian Foster, PhD
Director of Argonne's Data Science and Learning Division
University of Chicago

Chaofeng Wu
Research Assistant,
University of Chicago

Kyle Chard, PhD
Research Assistant Professor
University of Chicago

Zhuozhao Li, PhD
Postdoctoral Scholar,
University of Chicago

Ben Clifford
Software Developer
(Parsl project)