Modeling the computing requirements and costs for genomics analysis in the cloud

Michael C. Schatz[1,2], Bridget Carr[2], Victor Wen[2], Peiyuan Xu[2], Alex Mahmoud[1], Nuwan Goonasekera[3], Keith Suderman[1], Dannon Baker[1], Sergey Golitsynskiy[1], Kai Kammers[4], Sarah Wheelan[4], Stephen Mosher[1], Frederick J. Tan[5], Jeffrey T. Leek[6], Enis Afgan[1]

[1]Department of Biology, Johns Hopkins University, Baltimore, MD. [2]Department of Computer Science, Johns Hopkins University, Baltimore, MD. [3]University of Melbourne, Melbourne, Australia. [4]Department of Oncology, Johns Hopkins University, Baltimore, MD. [5]Department of Embryology, Carnegie Institution, Baltimore, MD. [6]Department of Biostatistics, Johns Hopkins University, Baltimore, MD.

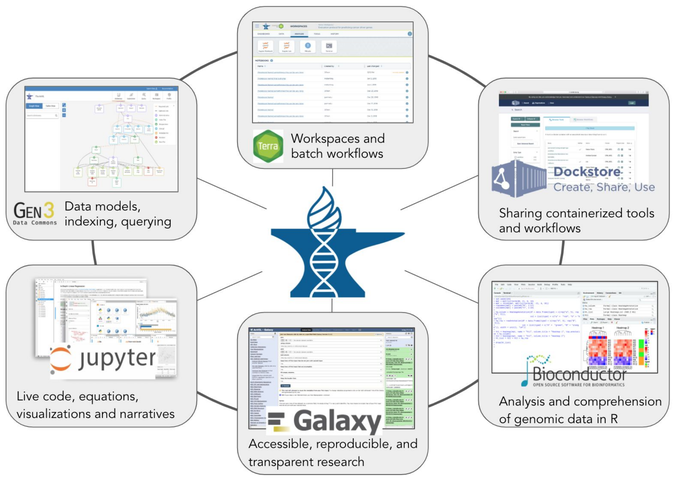
http://anvilproject.org http://usegalaxy.org

Abstract

Cloud computing offers many advantages for genomics analysis including security, scabilty, and simplicity. For example, the NHGRI Genomic Data Science Analysis, Visualization, and Informatics Lab-space (http://anvilproject.org) houses the genomics data for more than 300,000 samples, and offers thousands of analysis tools for interactive and batch computing in a FedRAMP-certified secure cloud computing environment. However, understanding and managing costs remain as some of the largest barriers for migrating biomedical related analysis into the cloud. Researchers currently have limited information for the expected costs for running analysis tools, which challenges budgeting and prevents many researchers from adopting cloud solutions. In addition, software developers may not focus on optimizing costs for cloud environments, which increases expense even when relatively simple optimizations are available.

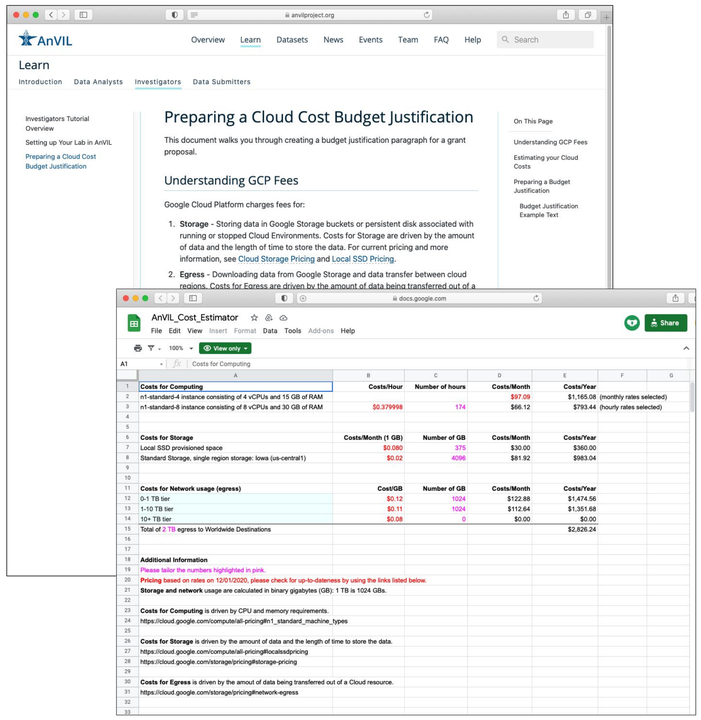
Addressing these needs, we are profiling and analyzing the cloud costs for many of the most widely used tools & workflows in genomics. To identify these workflows, we have mined the historic usage data on the global usegalaxy.* Galaxy servers, as it is one of the most popular community resources available. We are now measuring their computing requirements and costs when running with inputs of varying sizes and complexities. From these data, we aim to develop a predictive model and API that can estimate the costs for running these analyses on each of the major cloud platforms. Our goal is to inform investigators of the anticipated costs for their research and reduce costs by informing software developers of the tools that most urgently need optimization.

Why AnVIL?



The AnVIL is a highly scalable cloud analysis platform and data repository for genomics and biomedical research. On AnVIL, the Terra platform provides a compute environment with secure data and analysis sharing capabilities. Dockstore provides standard-based sharing of containerized tools and workflows. The Gen3 data commons framework provides data and metadata ingestion, querying, and organization. Jupyter, R/Bioconductor, and Galaxy provide environments for users to construct and execute analyses using a diverse set of tools for basic and clinical research. AnVIL provides access to many key NHGRI datasets with more than 300,000 samples currently available, including the CCDG (Centers for Common Disease Genomics), CMG (Centers for Mendelian Genomics), eMERGE (Electronic Medical Records and Genomics), GTEx v8 (Genotype-Tissue Expression Project), as well as other relevant datasets.

Cloud Budgeting



Computing Principles

- Your spend rate is primarily determined by the #virtual machines (cores x RAM x GPUs x disk) running in parallel plus the amount of cloud storage used
- Interactive analyses (e.g. RStudio or Jupyter notebooks) tend to be very inexpensive (<$1/hr) and will feel very familiar to desktop counterparts
- Batch analyses (e.g. WDL/CWL/Galaxy Workflows) vary enormously in computing costs from <<$1 to >>$10k
- Your spend rate is ultimately limited by your quotas.
  - GCP has separate quotas for VMs, cores, RAM, GPUs, IP Addresses, etc.
  - Increase quotas to accelerate
  - Decrease quotas to throttle spend
  - Benchmark, benchmark, benchmark….

Storage Principles

- Keep all essential input and output files to ensure work is reproducible
- Purge intermediate files after successful runs to limit long term data footprint.
- Collect metadata early and often; prefer existing ontologies and standards over developing custom formats
- Prefer compressed formats for long term storage, e.g. BAM or bgzip over fastq/SAM or vcf/txt/fa
- Some lossy formats may be acceptable, e.g. CRAM over BAM
- Cloud platforms may be able to provide”free storage” for certain shared datasets

https://anvilproject.org/learn/investigators/budget-templates

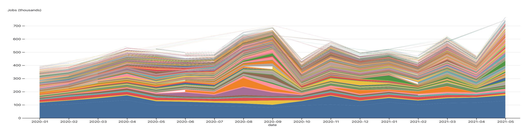
Benchmarking and Modeling

Phase I: Identify popular tools from historic usage at usegalaxy.org

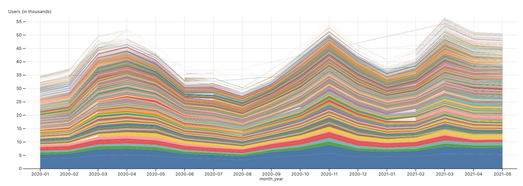
To identify the most widely used tools and workflows in genomics, we are analyzing the execution history within Galaxy for the past 3 years. By analyzing the Galaxy job execution history, we are developing an unbiased ranking across tens of thousands of active users, millions of analysis tasks, and thousands of tools that are currently available. This analysis leverages the extensive metadata that is tracked within the public instances of Galaxy, which records the input parameters (tool name, tool version, command line options, input files) of every job within a structured database. The raw execution data is summarized to aggregate across tool versions and across individual tool executions into workflows of tools that are executed in sequential order.

1. **Resource consumption**

2. **Number of jobs**



3. **Number of users**

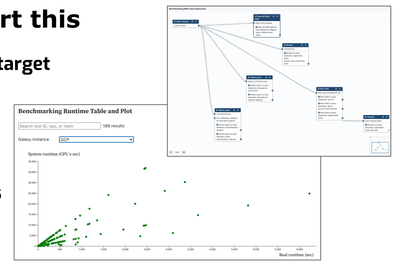


Phase 2: Benchmark and Modeling

We are now developing reproducible workflows to measure the computing requirements for the most widely used tools in Galaxy using prototypical datasets of different sizes. During the profiling, we are leveraging open-access data so that all of our results can be easily reproduced and extended by others. During these analyses, we are varying the number of CPUs, measuring the CPU time, peak RAM, and storage requirements on both the Google Cloud Platform and Amazon Web Services.

Components we built to support this

- **ABM: a script library for automating multi-target benchmarking**
- **ObservableHQ dashboard**
- **Benchmarking workflows**
- **Configurations for running the benchmarks**
- **API spec**



While we expect the computing requirements for many tools will scale linearly with the input size (the analysis of 10 genomes requires 10 times as much CPU time or costs as one genome), we are particularly interested to identify any tools that have super-linear requirements (e.g. an all-vs-all comparison of 10 genomes will naively be ~100 times as expensive as comparing a single pair). The modeling will use classic statistics (linear regression and low degree polynomials) so that the results will be highly interpretable and easily projected to datasets of any size.