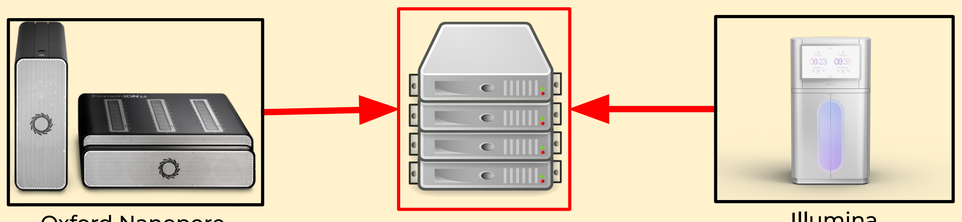
Hybrid Cloud and Local Data Lifecycle Management at a Medium-Scale Genomics Center

Sean M.L. Griffith, Benjamin Myers, Alice Sanchez, Michelle Mawhinney, Justin Paschall,
Beth Marosy, David Mohr, Elvin Hsu, Kimberly F. Doheny

Department of Genetic Medicine // Center for Inherited Disease Research (CIDR) // Johns Hopkins Genomics (JHG)
Johns Hopkins University School of Medicine, Baltimore, MD, USA

Obstacles in Data Management

The latest crop of sequencing technologies produce even more data.



Oxford Nanopore PromethION 24

"Storage is full again!"
- Crying IT Person

Illumina NovaSeq X Plus

All organizations have an increasing need for **recovery capabilities**, such as following a cyber-attack or local disaster.

"I can't sleep anymore!"
- Crying IT Person

An Ontology for Data Relationships

How is a set of genomics data constituted?
We have more than just the raw instrument data to consider.

Instrument Data → **Genomics Project**

Genomic Data

The end result of a project.

Data that will be released and used by investigators.

Examples file types include GVCF, BAM, CRAM.

Perpetual Project Data

These files tell the story of a project from initiation to release.

Typically composed of many small files,

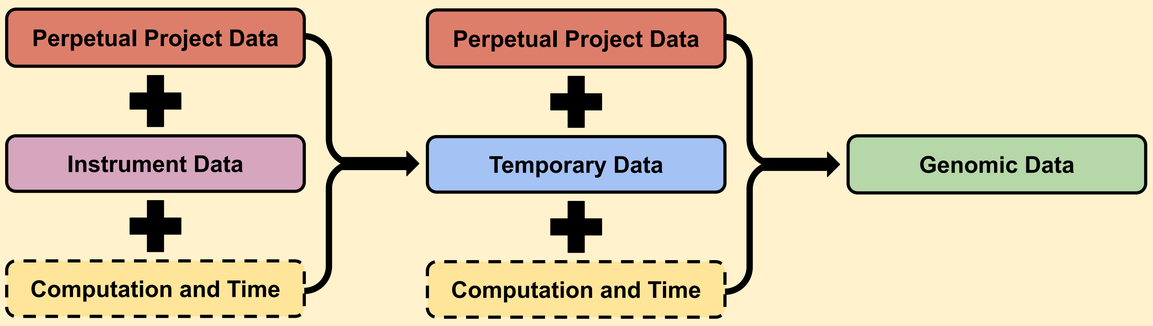
Often referred to after project release.

Temporary Data

Intermediate and ancillary files typically generated as genomic data are created.

Often occupy significant amounts of local storage.

What is the relationship among these data?



Given computation and time, perpetual project data and instrument data can be used to create genomic data.

How does this help?

These relationships guide what data needs to be replicated and how long data needs to be kept, informing how we should approach this problem.

A Solution: Targeted Cloud Storage

To overcome the data deluge, we need **more capacity** and to ensure recovery, we need **data replication**. Cloud storage solves both problems, but, engaged naively, significantly raises the costs of storing data.

An Additional Assumption:

We should prepare for emergency situations, but they are unlikely. Replicated data will rarely need to be used for full recovery.

What is the solution?

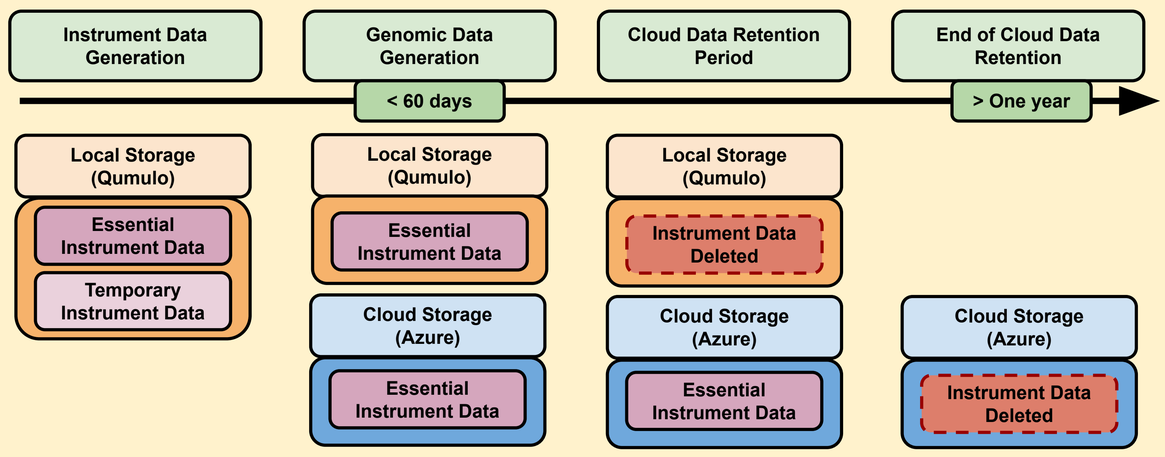
- Store all cloud data in the 'deep freeze' or archival tier. This storage is much cheaper, but more expensive to retrieve, due to thawing costs.
- For active projects, replicate instrument data and perpetual project data, which are the minimal data set needed to reconstitute those projects in the case of data loss.
- For completed projects, replicate the final genomic data, along with perpetual project data, keeping essential and useful data available.
- Most importantly, develop, follow, and automate data lifecycles that apply to both instrument data, all types of projects, and any other regularly generated bulky data.

Technical Details - The BlobLobber Suite

All local data are stored on a dedicated Qumulo NAS, with cloud data written to Microsoft Azure Blob Storage. Data transfer from local storage to the cloud is automated via BlobLobber, a custom suite of tools built around the Azure CLI using a combination of shell scripts, Java, and Python, with an accompanying back-end housed in a MySQL database. BlobLobber is designed to transfer data from local storage to the cloud in a massively parallel fashion, using available HPC resources during periods of low occupancy.

Example Data Lifecycles

Instrument Data



Project Data