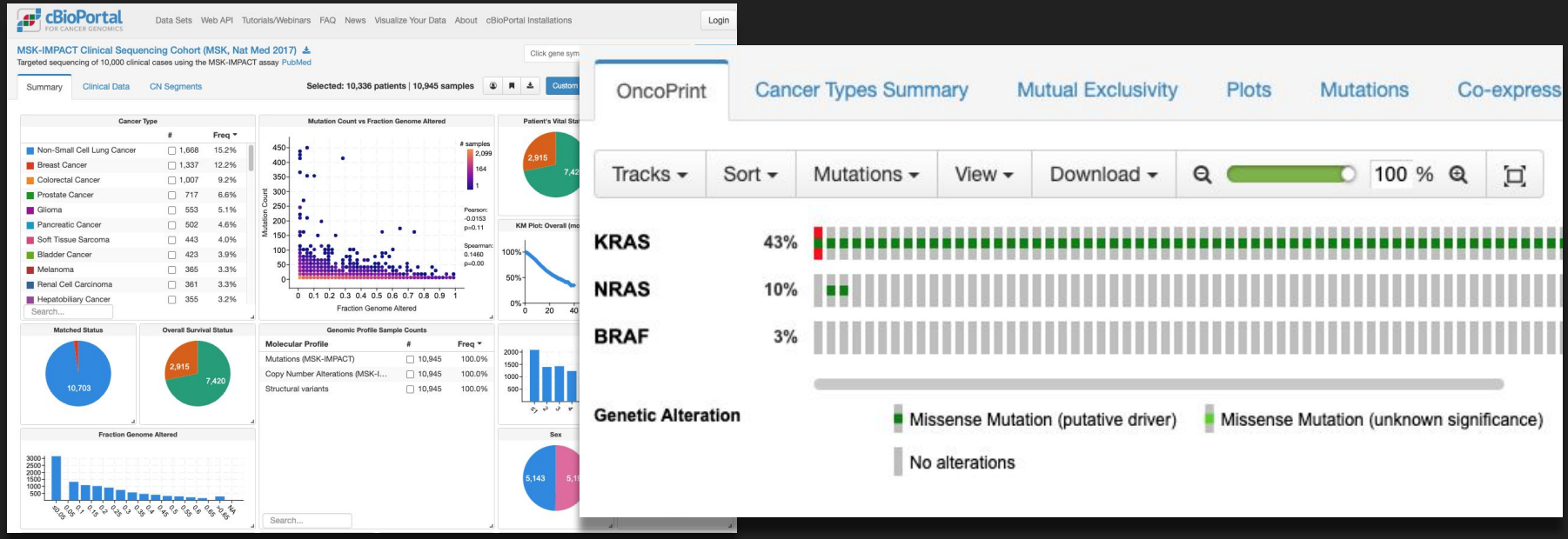
Cancer Research Made Faster

Who am I?

Aaron Lisman
Lead software engineer on the cBioPortal
Nikolaus Schultz Bioinformatics Lab
Center for Molecular Oncology at
Memorial Sloan Kettering Hospital

What is cBioPortal?

A data analysis and visualization application for exploring genomic and clinical data aggr from research studies and the clinic.



Origin of cBioPortal

- Originated in 2012 at MSK
- Open source
- Grant-supported
- Multi-institutional
- Used all over the world by tens of thousands of researchers and clinicians

History

1 year and 218 pull requests ago …

What is cBioPortal?

Bioinformatics for dummies

- Bioinformatics is data science applied to biological systems
- Draw conclusions that can inform treatment decisions and spark medical research based on patterns found in data.

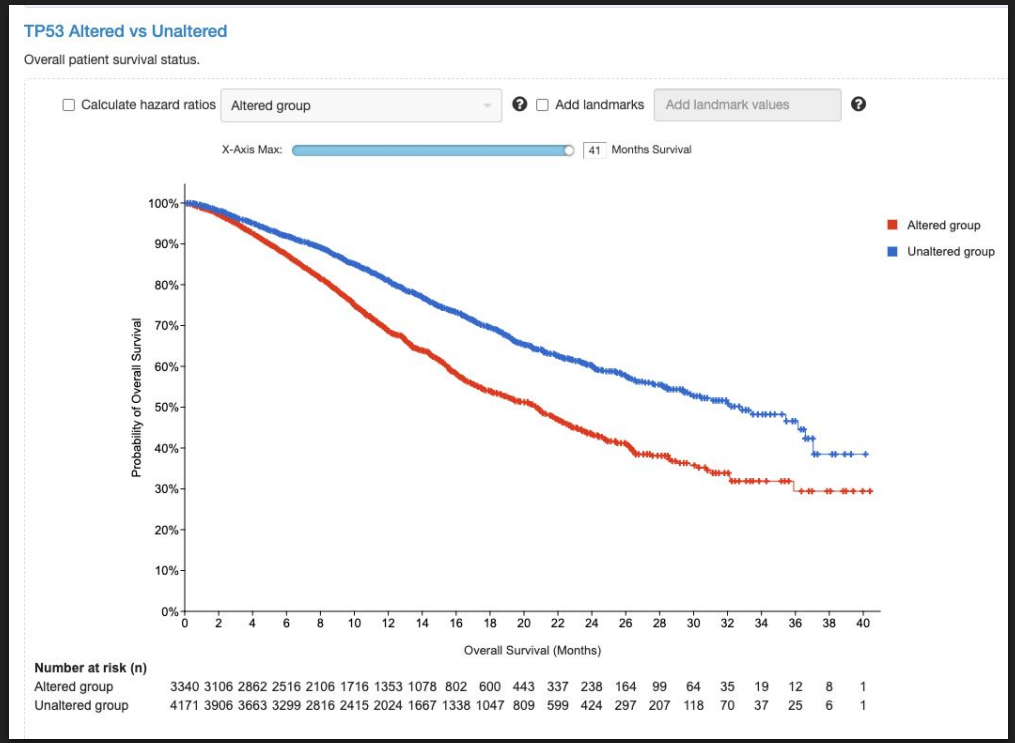
Cancer, a data-oriented disease

- A set of diseases
- Driven by a wide variety of genetic mutations
- Interfere with the body's ability to control tissue growth, i.e. tumors

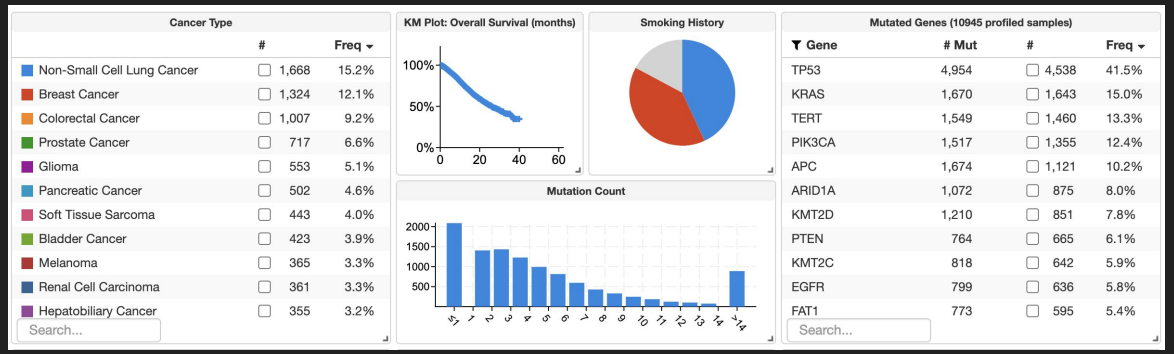
Decoding the mystery of cancer

- Relatively new ability to cheaply sequence DNA and detect a tumor's mutations
- Sequencing is almost standard-of-care now
- Lets us to peer into the root cause of cancer
- Suggests biological pathways that are involved and worthy of inquiry
- Target therapies at specific mutations

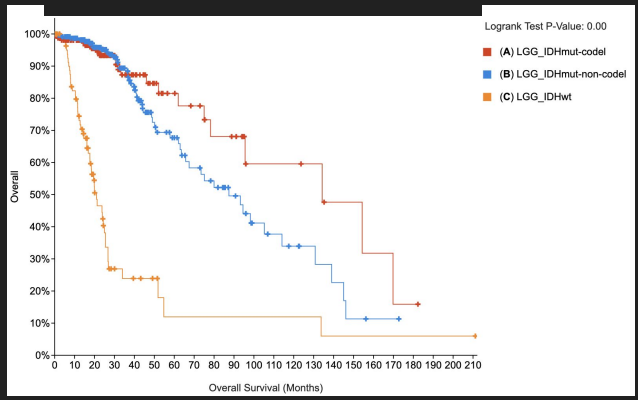
Test Hypotheses



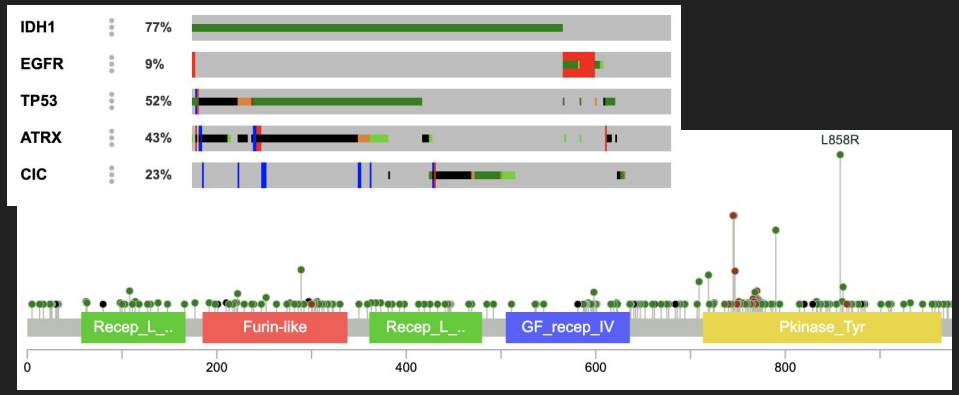
Study view: Cohort exploration



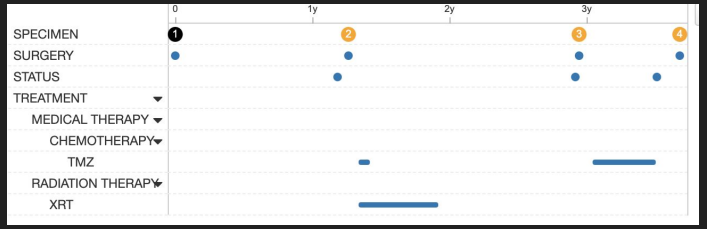
Group comparison



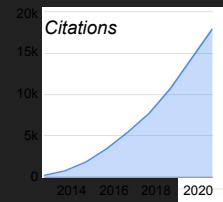
Results View: Gene-centric queries



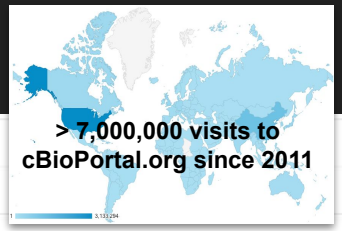
Patient View: Genomic and clinical timeline

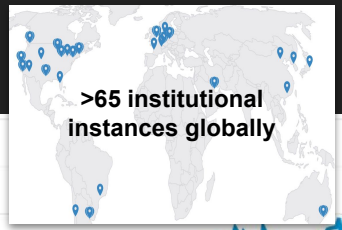


Samples	Gene	Protein Change	Annotation
1 2 3 4	IDH1	*R132C*	
1 2 3 4	TP53	*R248Q*	
4	ERC1	*L283Ffs*20*	
4	HSP90AB1	*K72E*	
4	SPRTN	*F404Lfs*3*	
1	OR10V1	*R273W*	
1 2 3 4	TEAD3	*P51L*	



Curated Studies	325
Monthly Users	~34k
Total Citations	~18k





Start of cBioPortal
Query/Results view

Patient View
Study View

Open source transition
(MSK, DFCI & Hyve)

PMCC &
CHOP joined

Architecture upgraded
Comparison View

Start of TCGA

First SU2C
cBioPortal

Cerami et al.
Cancer Discov.

Gao et al.
Sci Signal.
cBioPortal @MSK

cBioPortal @PMCC

cBioPortal @DFCI

Project GENIE

cBioPortal @CHOP

ITCR funding

Start of HTAN

Count Me In

GENIE BPC

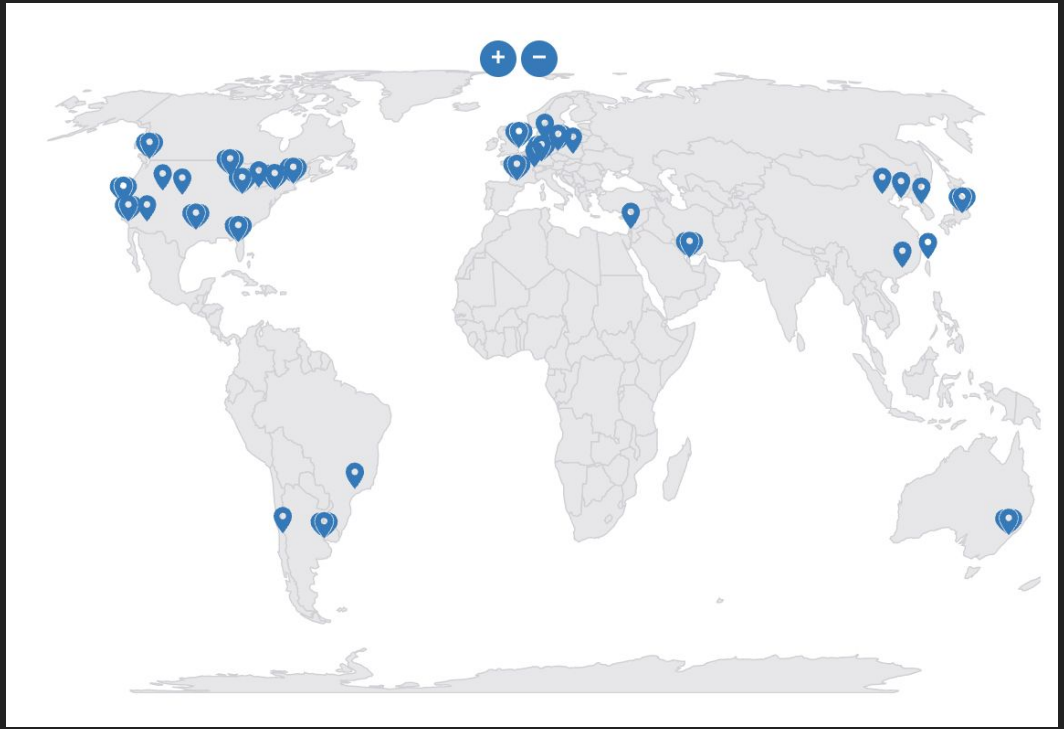
Webinars

cBioPortal FOR CANCER GENOMICS

https://cbioportal.org

@cbioportal

Hundreds of private instances worldwide



More performance = More insight

What is the scale of the data?

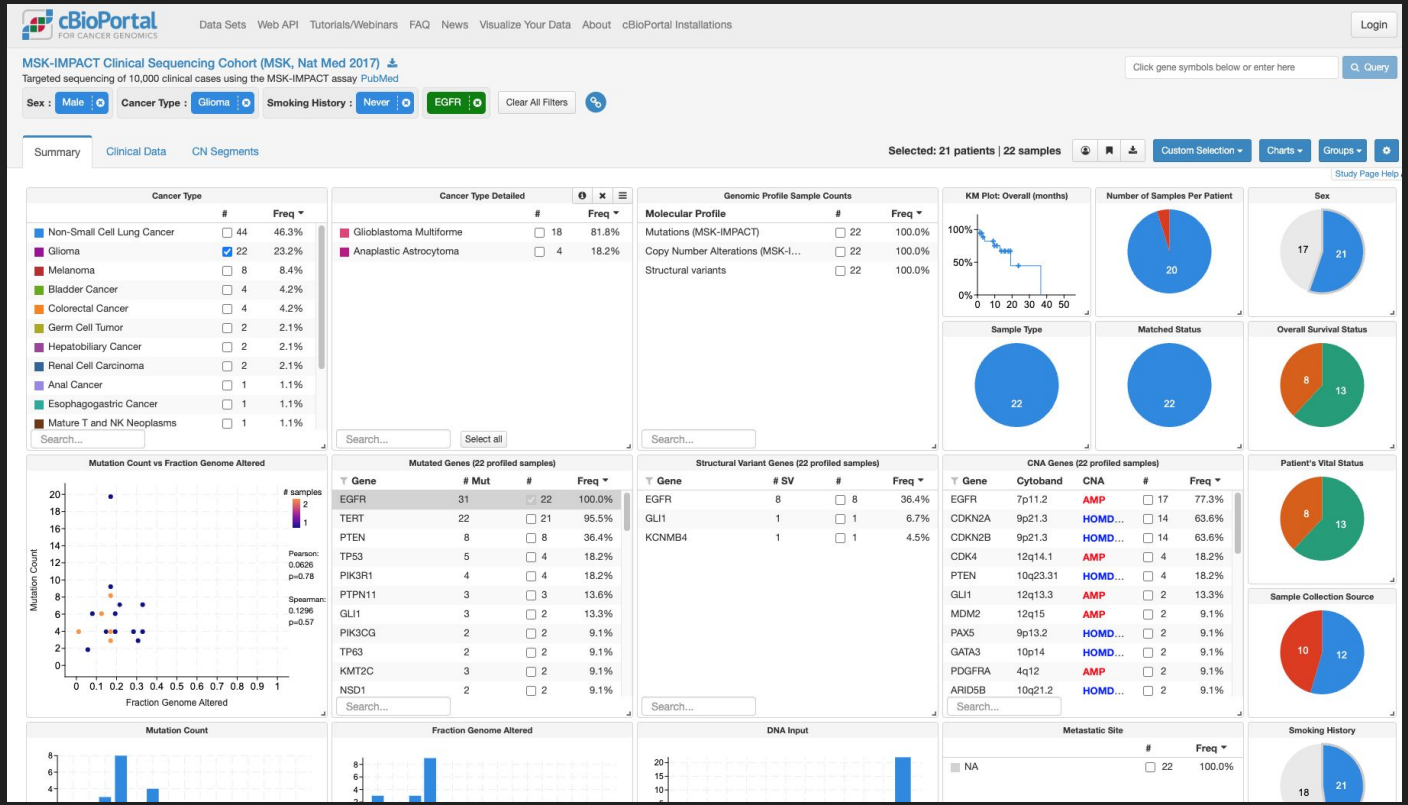
- Internal MSK Internal Portal has 140k patients
- Genie consortium: 220k

What is the scale of the data?

- The human genome has ~22,000 genes
- RNA expression data has a read per gene per sample
- = 4 billion rows per assay
- Goal: We want to support 1 million patients with multiple samples

Molasses

What are we actually doing?



OLAP!

Problem and strategy

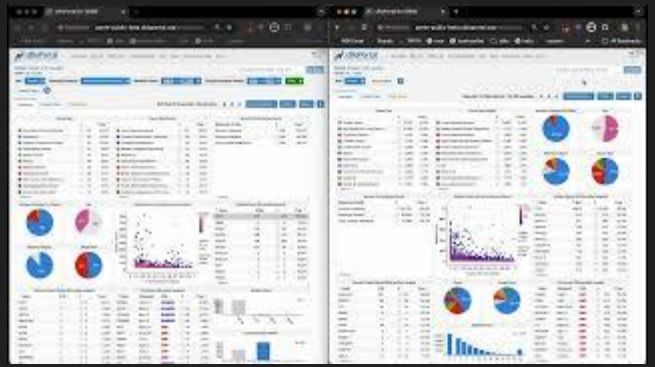
- We were doing much too much work in our service layer
- Bringing giant data sets into memory just to filter and count them in Java
- Exchanging performance for “developer ergonomics.”

6 month refactor

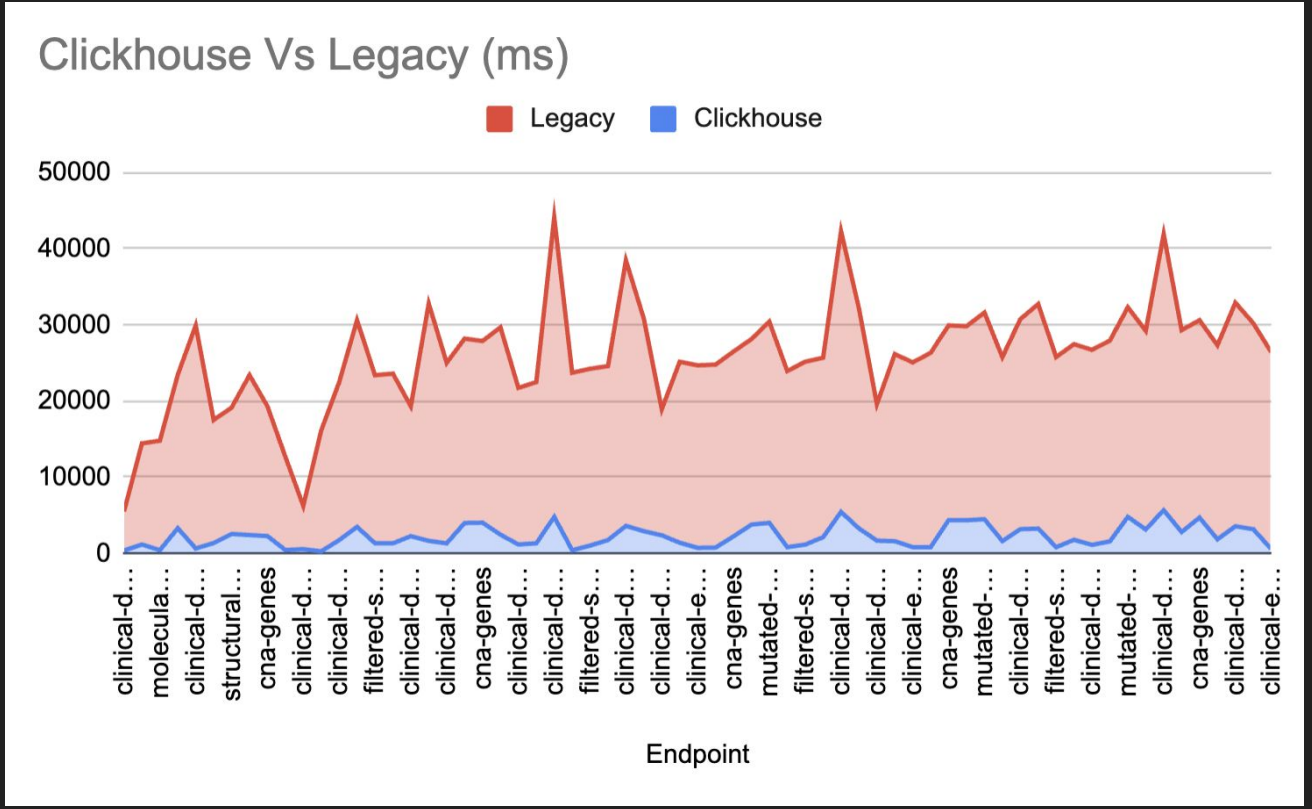
- Rebuilt 20 endpoints that filter patients and samples
- Built denormalized schema in Clickhouse according to the needs of these endpoints
- Reimplemented filtering logic in SQL

Success!

200k samples



Performance Improvements



~10x faster!

A	B	C	D
Endpoint	Clickhouse	Legacy	% Improvement
clinical-data-cou	393	5131	92.34%
filtered-samples	1181	13278	91.11%
molecular-profile	441	14368	96.93%
clinical-data-bin-	3320	20158	83.53%
clinical-data-cou	663	29278	97.74%
clinical-data-den	1373	16161	91.50%
structuralvariant-	2564	16586	84.54%
mutated-genes	2424	20998	88.46%
cna-genes	2294	17036	86.53%
sample-lists-cou	459	12243	96.25%
clinical-data-cou	571	5684	89.95%
clinical-event-typ	287	15865	98.19%
clinical-data-bin-	1754	20718	91.53%
clinical-data-bin-	3495	27127	87.12%
filtered-samples	1350	22057	93.88%
molecular-profile	1341	22274	93.98%
clinical-data-cou	2272	17056	86.68%
clinical-data-cou	1660	31153	94.67%
clinical-data-den	1333	23698	94.38%
structuralvariant-	4003	24207	83.46%
cna-genes	4047	23849	83.03%
mutated-genes	2473	27219	90.91%
clinical-data-bin-	1187	20562	94.23%
sample-lists-cou	1340	21159	93.67%
clinical-data-bin-	4804	39306	87.78%

ETL/Schema

- Wanted to get right to business proving the optimization concept
- Copied the MySQL schema whole into Clickhouse using Sling
- Derive denormalized views based on the underlying tables
- Materialized view issues when based on complex joins

Nests of nests

- Mantra was, do not return voluminous data to the web server.
- Forces you to get creative and complicated with subqueries.

Logic in database

Much business logic now in form of complicated, deeply nested SQL

```
        Otherwise - The table can be filtered on both patient id(s) and sample id(s)
     -->
    <sql id="applyStudyViewFilter">
        <choose>
            <when test="${filter_type} == 'PATIENT_ID_ONLY'">
                <include refid="applyStudyViewFilterUsingPatientId"/>
            </when>
            <otherwise>
                sample_unique_id IN ( <include refid="sampleUniqueIdsFromStudyViewFilter"/>)
            </otherwise>
        </choose>
    </sql>
```

- MyBatis helps with modularity
- Still hard to reason about and debug
- Unit testing very difficult

Finish the job

Now that we've proven the optimization works:

- Can the rest of the app's less OLAP-oriented functionality perform sufficiently using existing legacy SQL running against Clickhouse's MySQL interface?
- Can we use one database, or do we need both?

All-Clickhouse?

- Single database is attractive
- Do we denormalize everything in the the ETL process?
- How do we maintain the data integrity checks provided by a conventional relational schema?

Remaining technical issues

- Complicated custom binning logic for histograms
- Still using legacy approach
- Can this be accomplished in database?
- User defined functions?

Lots of Success
Lots more Work!

Thank you Clickhouse!