omero as a general purpose framework for biomedical data management
Our first goals (about 3 years ago)

• to have scalable, uniform, computational access to large amounts of *-omic heterogeneous data
  – From bio-samples to next gen sequencing data
• to be able to track data dependencies
  – model both objects and actions that connect them
• to support computation on meta information and data dependency tree
  – E.g., plan optimal titer-plate loading for next experiment
• to support data access from multiple, geographically distributed, labs
  – {Pula,Alghero,Lanusei,Monserrato}@sardinia, ...
• but first and foremost: no more excel sheet (!)
**omero.biobank**

- specialization of the "omero framework" to the handling of *omic data
  - customized models and data structures for biomedical data handling: Genotyping data, clinical records, vessels, ... (49 customized models)
  - network of objects connected by actions
  - can track transformations performed on the data
  - provides a rich API and tools for data input and queries

- **heavy use of omero tables**
  - snp markers, markers set, alignments, phenotype records

- **all client side code (~30k lines)**
  - mostly syntactic sugar
  - mostly boring stuff (importers/exporters/...)

omo. biobank: use

• Data mainly from two large scale studies
  – autoimmune disease (CNR-IRGB)
  – longevity (CNR-IRGB, NIH-NIA)

• Currently handling:
  – > 38000 individuals (~16.500 with parental relationships)
  – 26.800 clinical records
  – ~28.200 vessels, ~330 Titer Plates
  – 4 Genotyping technologies
    • Affymetrix GWH 6.0 (~935.000 markers, ~7.000 gtypes)
    • Illumina Immunochip (~196.000 markers, ~10.000 gtypes)
    • Illumina Hu OmniExpress (~730.000 markers, ~3.000 gtypes)
    • Illumina Hu Exome (~ 240.000 markers, ~5.000 gtypes)
omero.biobank: problems

• Not particularly biologist-friendly
  – Programmatic/script interface too complex for casual user
  – Tracking complex operations (action(s)) is rather cumbersome

• Need to access multiple computing environments
  – Batch system
  – Hadoop
    • largest cluster 3200 cores, uses an 'elastic' hadoop-grid-engine resource allocation scheme
  – Different filesystems

• Users are in different locations:
  – From the same island to different continents
**omero.biobank: omero specific problems**

- **no omero integrated solution for dependency graph navigation**
  - We are currently using client side solution (pygraph) [slow]
  - Next: external graph handling service [fast, but dangerous]

- **slow on large data (tables) operations**
  - improved with ColumnArray<X>
  - more on this later

- **external file handling headaches**
  - DataObjects point to physical files not directly managed by omero
refined goals (18 months ago)

• to have a simple, biologist friendly, user interface
• to simplify standard data processing
  – facade to hadoop, batch job submission
• tools to build and share workflows
• maintain history of operations performed
  – share histories, save histories in omero,…
• decouple logical file view from file system details
  – meta-information based file system
omero.biobank + galaxy + iRODS
Galaxy (usegalaxy.org) web interface for CLI tools

History of operations performed
Galaxy: quasi-lab-book
Galaxy: workflow editor
Interaction with omero.biobank

Galaxy / CRS4 IRGB

VLT.select_sub_group (version 1.0.0)

**study label:**
Use all individuals
Select only between individuals enrolled in this study. See below.

**Group label:**
fake-group
the new group (it is actually a study) label

**total number of individuals requested:**
100
It will be cut to the largest number of individuals satisfying the required constraints.

**Male fraction:**
0.5
The fraction of male individuals.

**Phenotypic profile (diagnosis, for the time being):**
Type 1 Diabetes

**Control fraction:**
0.5
The fraction of control individuals.

**Required datasample type:**
none
Façade to hadoop tools
iRODS as a Decoupling System

• **IRODS** is an integrated Rule-Oriented Data-management System
  - uses unique logical names that are separate from the names as stored physically, providing a global ‘logical name-space’
  - Rules to automatically treat data on insertion and retrieval
  - Ability to tag data sets (e.g., sample id, data format)
  - Web based and command line interfaces
  - transfers data across the network in an integrated manner (parallel threads for large files)

• **We use IRODS as a front end to our heterogeneous storage system**
  - about 4.5PB in various boxes

**iRODS is developed by DICE UNC** ([http://www.irods.org](http://www.irods.org))
Short-term vs long term memory

- **Typical workflows**
  - have several steps and may fail
  - unwise to commit intermediate data to repository

- **Solution:**
  - Short-term memory → Galaxy history
    - Tracks steps while the computation is running
    - Permits to iteratively build a “good protocol”
  - Long term memory → OMERO.biobank
    - Record history in OMERO.biobank
**galaxy + omero + iRods: glue**

- **extensions to galaxy**
  - support communication with omero.biobank
  - improved galaxy histories API to support omero consumption
  - Almost all relevant tools galaxy wrapped
    - omero.biobank import/export/query tools
    - hadoop based tools for NGS and genotyping
    - ....
  - we are extending galaxy objectstore to directly support iRODS objects (files and collections)

- **iRODS**
  - external reference data is moving to iRODS
  - omero.biobank is moving to irods:// file paths
  - iRODS rules to simplify registration of huge dataset and galaxy integration
galaxy + omero + iRods

- **User community: biologist/bioinformaticians**
  - About 50 external, 10 internal users
  - All omero.biobank import, most export and queries

- **Problems:**
  - «designed» to have a human in command
    - Manage complex workflows chains, handle failures
  - Boring, dangerous and expensive for large scale production runs
new goals (5 months ago)

- support the running of the CRS4 next generation sequencing service (3 Hiseq-2000)
  - From biological sample in the mail to digital data in the cloud
  - automatize anything that would be cost-effective to automatize
Yet another full Data cycle Automator
Automation

• Galaxy front-end for biosample submission and analysis request
• All data operations described as galaxy workflows
• Automation layer that chains together workflows and integrates the various system components:
  – Illumina sequencers
  – Galaxy (-> Hadoop cluster)
  – omero.biobank
  – iRODS
• Basic pipelines up and running
  – Flowcell to per-sample fastq datafiles in production
Sample submission front-end

Github fork of Brad Chapman (Harvard Med School) system
Big data workflow

iRODS
Integrated Rule-Oriented Data System

Pathsets
location meta-data

OME biobank

Galaxy

Actual operations on the data
to summarize: our mantra

- omero.biobank knows what things are
- iRods knows where things are
- galaxy knows how to operate on them
Back to one of our slowness problems

\[ R = \sum f(x) \]

Client side (current)

Server side (classic)

Server side (map-reduce)
Processing rates

Reading a pytable with a map-reduce program (16 nodes)

1.2Tbytes

Scalar and parallel processing rates are compared. The graph shows the processing rate in rows/sec on the y-axis against the logarithm of the number of rows on the x-axis. The data points indicate a linear increase in processing rate with increasing data size.
Structured objects file system

• Possible to instruct/delegate computing framework on how computational load should be distributed

• HDF5 natural candidate to impose «scientific data» structure on file system
  – Implementation details
    • using H5FD_SPLIT it is possible to separate data from metadata in two different files
    • In principle possible to have HDF5 on top of HDFS, QFS better?
    • We wrote a minor pytables extension to support H5FD_SPLIT, so we can easily try on HDFS (and later on QFS)

• BTW- For this class of objects, e.g., big SNP arrays, HBASE is not a good solution.
new goals: back to images!

• We are moving toward “pathology” applications support
  – Integration of sequencing + proteomics + digital pathology
THANK YOU FOR YOUR TIME!