(ab)using omero

Gianluigi Zanetti

(gianluigi.zanetti@crs4.it)
• Center for Research, Development, and Advanced Studies in Sardinia
• Interdisciplinary research center focused on computational sciences
• Located in the POLARIS Science and Technology Park (Pula, Sardinia, Italy)
• Operational since 1992
• RTD staff of ~180 people
Sardinia and its genetic isolated founder populations

- Sardinia is characterized by genetic isolated founder populations
  - An island that is big enough and where it is difficult to travel
- CRS4 cooperates with CNR-IRGB in two large scale studies
  - on longevity
  - and auto-immune diseases
Progenia: search for genetic influences on longevity

- Joint project CNR-IRGB/NIH
- Population level studies
  - 6,148 individuals - aged 14-102y, 95% are known to have all grandparents born in Sardinia
- Highly characterized samples
  - Traits (> 150) ranging from anthropometric measurements to personality facets, repeated measurements on the scale of a decade
  - High resolution genotyping, (soon) deep sequencing
- Cast of ‘00, see *

* http://sardinia.nia.nih.gov/Project_Team/project_team.html
Auto-immune diseases such as type-1 diabetes and MS have in Sardinia one of the highest incidences world-wide.

- Collaboration with CNR-IRGB
- Population level study
  - 4,000 affected individuals and 2,000 healthy volunteers
  - High resolution genotyping
  - Order of $10^2$ reseq.
- Cast of ‘00, see *

Scale of the problem (1/4)

- **genotyping**
  - Low cost
  - Many (10^4)
- **Deep sequencing**
  - Higher cost
  - Fewer (10^3)

High Res. Genomic Picture

Clinical information

Chen and Abecasis AJHG 2008
• Transition from independent ‘labs’ to a single distributed pipe-line
  • Biosamples are geographically distributed
  • Multiple genomic technologies
  • Multiple clinical data sources
    • Existing biobanks (CNR-IRGB, Progenia)
    • Feed from Regional Health system
  • Comparable with a small hospital

• Non trivial ‘Data intensive’ problem
  • Genotyping dataset order of $10^4$ ind.
  • Deep sequencing datasets order of $10^3$ ind. (>4TB/week)

• Moving Target
  • More detailed clinical data
  • Soon start adding epigenomic data, e.g., ChIP-seq, RNAseq, …
A software crisis

We are here

Processing capacity
Sequencing capacity
Example: re-sequencing pipeline (old way)

- Direct “transposition” of serial pipeline to GridEngine
  - Shared file system

- Low parallelism
- Job failures
- Shared storage failures
- Difficult maintenance
Example: re-sequencing pipeline (new way)

Scalable!
A software crisis: aka excel will break!

Distributed excel is an inherently brittle technology
Need to capture statics and dynamics

- Electronic Health Record
  - Multiple sources
  - Implementation specific details
- Samples
  - Bio and synthetic
    - Physical location / big dataset
  - Chain of Custody
- Operation description
  - ‘Experimental’ and ‘digital’ ops
- Computational driven inference process
  - Uniform access to data
Desiderata: Model state and transitions

Raw data sample

actions

Objects with state
Desiderata: Keep track of dependencies

Raw hybridization data

10,000

Data collection (and subsets thereof)

Genotyping results
### Desiderata:

**Uniform SNPs mngmt**

<table>
<thead>
<tr>
<th>AA</th>
<th>0.2</th>
<th>0.9</th>
<th>0.3</th>
<th>0.1</th>
<th>...</th>
<th>dataset1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BB</td>
<td>0.8</td>
<td>0.1</td>
<td>0.7</td>
<td>0.9</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AA</th>
<th>0.9</th>
<th>0.9</th>
<th>0.1</th>
<th>...</th>
<th>dataset2</th>
</tr>
</thead>
<tbody>
<tr>
<td>BB</td>
<td>0.1</td>
<td>0.1</td>
<td>0.9</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

**Reference genome**

**Flanking regions**

CCGACCTAGTTGGCAA[A/G]TAGACACTGAGGCTGA
def main():
    kb = KB(driver='omero')(...)
    maker, model = 'crs4-bl', 'taqman-foo'
    mset = kb.get_markers_set(maker, model)
    s = gkb.get_gdo_iterator(mset)
    counts = algo.count_homozygotes(s)
    mafs = algo.maf(counts)
    hwe  = algo.hwe(counts)
def main():
    kb = KB(driver='omero')(...)
    ...
    enrolled = kb.get_enrolled(study)
    #--
    for e in enrolled:
        dsets = kb.get_gdos(e.individual)
        support, mean, sigma = compare_dsets(dsets)
def compare_dsets(gdos):
    support, mapping = algo.find_shared_support(gdos)
    ad = [np.vstack([g['probs'][:,i], g['confs'][:,i]])
           for (g, i) in it.izip(gdos, mapping)]
    map(lambda _ : np.reshape(_, (1,) + _.shape), all_data)
    all_data = np.vstack(ad)
    v  = all_data[:,0:2,:].sum(axis=0)
    v2 = (all_data[:,0:2,:]**2).sum(axis=0)
    N = all_data.shape[0]
    mean = v/N
    # FIXME: I know, this is not the variance...
    var = v2/N - mean**2
    return (support, mean, np.sqrt(var))
Computable data semantics is crucial

"Semantic understanding belongs to the humans that wrote the adapter"

Gianluigi Zanetti / June 2011
Computable clinical semantics

... ADL SNIPPET
ELEMENT[at0078.7]
  occurrences matches {0..1} matches {
-- Mean cell haemoglobin conc. (MCHC)
  value matches {
    C_DV_QUANTITY <
      property = <[openehr::119]>
      list = <
        ["1"] = <
        units = "gm/l">
      magnitude = <|>=0.0|>
  ...

Keep biomedical & computational semantics

Object metadesc

Name: foo
Type: measured_value

e.g., openEHR ADL
Tech. Agnostic and future proof

<table>
<thead>
<tr>
<th>GRA</th>
<th>GRA%</th>
<th>HCT</th>
<th>HGB</th>
<th>LYM</th>
<th>LYM%</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.44</td>
<td>46.42</td>
<td>45.50</td>
<td>145.05</td>
<td>2.43</td>
<td>46.21</td>
</tr>
<tr>
<td>2.50</td>
<td>57.98</td>
<td>41.46</td>
<td>135.59</td>
<td>1.57</td>
<td>36.37</td>
</tr>
<tr>
<td>3.68</td>
<td>61.58</td>
<td>48.14</td>
<td>151.58</td>
<td>2.01</td>
<td>33.68</td>
</tr>
<tr>
<td>2.56</td>
<td>52.69</td>
<td>46.07</td>
<td>144.71</td>
<td>1.91</td>
<td>39.34</td>
</tr>
</tbody>
</table>
Why we like omero

Omero is agnostic
  Configurable, distributed, platform that deals with collections of objects
  Agnostic vs objects models
  Agnostic vs programming languages (client side)

Omero can grow
  Meta class description of objects
    Automatically map openEHR archetypes to models
  Omero Tables!
  Minimal down-time for model set extension
    Keep db, install delta (probably an hack)
Gianluigi Zanetti / June 2011

Misusing Omero

OMERO
(with unauthorized Models)

ICE middleware

Open Mirth
Demogr. GW (HL7v3)

Direct data collection

Clinical Data gathering

Regional Health System

onyx
onyx
onyx

MEDIR

Hospital Lab
Hospital Lab

DJANGO

Lab Sassari
Lab Lanusei
Lab M.serrato
Lab Pula

proto LIMS

web clients

SCRIPTS

Sample tracking
More Omero misuses

OMERO
(with unauthorized Models)

ICE middleware

Meta job dispatcher

Hadoop jobs
Desiderata: Keep track of dependencies

- Raw hybridization data
- Genotyping results
- Data collection (and subsets thereof)

10,000
An omero specialization
New models and related sugar

```python
class Device(wp.OmeroWrapper):
    OME_TABLE = 'Device'
    __fields__ = [('vid', wp.VID, wp.REQUIRED),
                  ('label', wp.STRING, wp.REQUIRED),
                  ('maker', wp.STRING, wp.REQUIRED),
                  ('model', wp.STRING, wp.REQUIRED),
                  ('release', wp.STRING, wp.REQUIRED),
                  ('physicalLocation', wp.STRING, wp.OPTIONAL)]
```

```xml
<type id="ome.model.v1.Device">
    <properties>
        <required name="vid" type="string" unique="true"/>
        <required name="label" type="string" unique="true"/>
        <required name="maker" type="string"/>
        <required name="model" type="string"/>
        <required name="release" type="string"/>
        <optional name="physicalLocation" type="string"/>
    </properties>
</type>
```
• Omero is much more than bioimages handling
• Adding sugar has been enough for our needs
  – There could be some ad-hoc improvements…
• We are hiring….