Supplementary Note

OMERO General Overview

OMERO is a client-server application for managing large, heterogeneous biological datasets. The overarching documentation site for OMERO is http://trac.openmicroscopy.org.uk/ome/wiki/OmeroHome. All software is available at http://openmicroscopy.org. Plans and roadmaps are publicly released at http://trac.openmicroscopy.org.uk/ome/roadmap. Comments, questions and feedback are welcome via email lists or on-line forums listed at http://www.openmicroscopy.org/site/community. Questions and feedback can be detailed questions on OMERO software or questions from users on how to use the software and accomplish a specific task. We aim to respond to all queries within one work day of posting (UK national holidays excluded).

Documentation for OMERO users is based at http://www.openmicroscopy.org/site/products/omero. Documentation for developers is based at http://trac.openmicroscopy.org.uk/ome/wiki/OmeroHome. Many pages covering specific issues are linked from these pages. For example, detail on how the OMERO uses code generation is at http://trac.openmicroscopy.org.uk/ome/wiki/ObjectModel.

OMERO is an open source development project. The software is available under the GNU GPL license (http://www.gnu.org/licenses/gpl.html). The teams participating in the project are listed at http://www.openmicroscopy.org/site/about/development-teams. The project is directly funded by grants from the Wellcome Trust and the BBSRC. As of December 2011, funding for the full Dundee-based development team and single developers associated with the University of Edinburgh, Imperial College London, Oxford University (UK), Institut Pasteur (France), CRS4 (Italy), University of Wisconsin, NIA-NIH, Carnegie-Mellon University, and Harvard Medical School (USA) has been secured through a Wellcome Trust Strategic Award through September 2014. Glencoe Software, Inc. makes significant contributions to the open source OMERO codebase.

The following sections highlight some features of OMERO and specific details of OMERO integration.

Permissions to Share Data

The policies for data access differ between research environments and indeed between neighboring laboratories. OMERO implements a mechanism for defining and using permissions to control data access and sharing. All data in an OMERO database are stored with an explicit declaration of permissions, and custom Hibernate filters limit the data returned to a client to match the permissions criteria. User/Group/World policies based on the standard Unix framework are the basis for these permissions. A root user (equivalent to a super-user or Administrator in Windows) is supported, as well as a 'group leader', who has root level privileges within a group. Users may have membership in multiple groups, which can be declared as private, collaborative-read only, or collaborative-read/write. Both collaborative states allow group members to see each other's data, but only groups declared as collaborative-read/write allow group members to annotate another user's data. A user and a group leader are the only ones that can delete a user's data. In practice, this approach lets a group of scientists define data they will and will not share, and where allowed, view and discuss each other's data, using standard remote applications (web browsers, desktop applications, etc.).

Some research environments require more flexibility for sharing than the system based on group membership provides. Therefore, OMERO provides a mechanism to increase the visibility of a user's own data beyond that defined in the permission system. Groups of images can be added to any number of "shares". Access to the share can be granted to scientists outside of a user's OMERO group(s) who can then view and take part in discussions about them.

Example Usage of OMERO.tables

The OMERO.tables API (http://trac.openmicroscopy.org.uk/ome/wiki/OmeroTables) unifies the storage of columnar data from various sources, such as automated analysis

results or script-based processing, and makes them available within OMERO. With this API, tabular (i.e. spreadsheet-like) data can be stored in and HDF5 file (http://www.hdfgroup.org/HDF5/) via named columns, and retrieved in bulk or via paging. This is made available using a PyTables interface (http://www.pytables.org). A limited query language provides basic filtering and selecting. To store and query measurements of counts of nuclei per well at specific cell cycle stages:

1. create the table columns

```
iId = ImageColumn('Image_id', 'Image ID', list())
nucCount = DoubleColumn('Nuclear_Count, '', list())
stage = StringColumn('cell_cycle_stage', 'Cell cycle stage', list())
```

2. populate values

```
iId.values.append(100L)
nucCount.values.append(47)
stage.values.append("Metaphase")
```

3. create the table

```
columns = [iId, nucCount, stage]
table = session.sharedResources().newTable(1,'/mytable.h5')
table.initialize(columns)
hdfFile = table.getOriginalFile()
```

4. query the table

```
array = table.readWhere('(nucCount < 50) && (cell_cycle_stage  == "Metaphase")')
```

Integration with VolViewer

VolViewer (http://www.uea.ac.uk/cmp/research/cmpbio/VolViewer) was made to interface with OMERO for reading images. This was achieved by using the OMERO C++ library and interfacing with the OMERO Gateway service (http://trac.openmicroscopy.org.uk/ome/wiki/OmeroCpp). This allowed VolViewer to read image metadata and binary data directly from an OMERO server. More details about the exact implementation, including source code and how to build VolViewer with OMERO support can be found on the VolViewer project page (http://dmbi.nbi.bbsrc.ac.uk).

Links with Third Party Image Processing and Analysis Software

See the following URLs for more information and documentation on linkages to external image processing applications:

ImageJ: OMERO.ij; http://openmicroscopy.org/site/support/omero4/downloads

CellProfiler: http://trac.openmicroscopy.org.uk/omero/wiki/OmeroCellProfiler

Matlab: http://trac.openmicroscopy.org.uk/omero/wiki/OmeroMatlab

OMERO-based Image Repositories

OMERO has been used as the foundation for two public repositories, the JCB DataViewer (http://jcb-dataviewer.rupress.org) and the ASCB's CELL Library (http://cellimagelibrary.org). We have also reworked an existing image data repository, the Electron Microscopy Data Bank[5] (EMDB; http://www.emdatabank.org) using OMERO (http://emdb.openmicroscopy.org.uk).

Using OMERO as a Foundation for Genotype Analysis

The GWAS version of OMERO uses OMERO.tables to handle large sets of clinical electronic health records and associated genotyping datasets, including SNP marker definition, SNP alignment to reference genomes, the grouping of markers for different genotyping technologies, and the final imputed genotype data stored as arrays of probabilities. By adopting this general storage approach, it becomes easy to access data derived from different genotyping technologies including high-throughput platforms as well as other approaches for lower numbers of assays but high sample throughput such as TaqMan SNP genotyping assays and the results of the analysis of high throughput resequencing data[1]. Most importantly, these data can be processed very efficiently, since many of the basic analysis, e.g., maf, Hardy Weinberg exact calculation[2] and other data quality controls, can be expressed as parallel/streaming operations across rows in the OMERO.tables architecture and be performed very close to I/O bandwidth speed. Alternative approaches (e.g., GenoByte; http://www.obiba.org/node/75) use a compressed bit representation of the SNP call that reduce data volume, but do not retain a measurement of uncertainty for each SNP.

In our implementation, the computational parameters and submission plan for the imputation jobs are computed first, followed by the actual, batch jobs that, access the metadata and the specific datasets contained in OMERO. Input data files for the imputation

runs, with pedigree and genotyping information, are built on the fly in the QTDT format (http://www.sph.umich.edu/csg/abecasis/QTDT/docs/ibd.html), and then submitted to a specialized, custom built, Hadoop application that manages the distribution of the related jobs to the computational clusters. This approach means that processing metadata and setting up a calculation takes minutes at most, whereas directly reading metadata from the raw data files (ped files, ~20 GB each) takes many hours.

Genotype Calling (GC)[3] represents an important challenge for GWAS, given the noise associated with high-throughput genotyping and batch effects when performing plate-by-plate processing. In order to reduce batch effects and other errors[4], as many samples as possible must be processed together and systematic analysis of the dependence of the results on the specific dataset used must be determined. However, as the number of samples grows, job run time and computational resource usage increase dramatically, making it difficult to scale efficiently to thousands of samples. Even partitioning data into separate probesets and allowing a certain level of parallelization, does not solve the problem, since currently available algorithms require access to all data samples at once. For instance, a GC run performed at our site using the Affymetrix Power Tools (APT), took two weeks for processing 6863 samples by using 18 nodes with 8 CPUs and 16 GB of RAM for each job (one node for each probeset). These computational costs make it cumbersome to perform a systematic analysis of the dependence of result robustness on the dataset used.

We developed a new distributed GC application that scales efficiently with the number of samples, harnessing the full power of a large computational cluster. For example, we were able to perform GC on the aforementioned 6863 samples in about 16 hours using 30 nodes with 8 CPUs each. We use OMERO to generate groups of homogeneous (e.g., gender and phenotype) data sample collections that are then used to perform the analysis and investigate its robustness; we store the results in the OMERO DB or OMERO.tables, and then use OMERO to simplify the scripting of the subsequent processing. The modelling for this system is based on OpenEHR archetypes (http://www.openehr.org).

References

1. Nielsen, R., Paul, J.S., Albrechtsen, A. & Song, Y.S. Genotype and SNP calling from next-generation sequencing data. *Nat Rev Genet* **12**, 443-451 (2011).
2. Wigginton, J.E., Cutler, D.J. & Abecasis, G.R. A Note on Exact Tests of Hardy-Weinberg Equilibrium. *The American Journal of Human Genetics* **76**, 887-893 (2005).
3. Cantor, R.M., Lange, K. & Sinsheimer, J.S. Prioritizing GWAS results: A review of statistical methods and recommendations for their application. *American journal of human genetics* **86**, 6-22 (2010).
4. Hong, H. *et al.* Assessing sources of inconsistencies in genotypes and their effects on genome-wide association studies with HapMap samples. *The pharmacogenomics journal* **10**, 364-74 (2010).
5. Lawson, C.L. *et al.* EMDataBank.org: unified data resource for CryoEM. *Nucleic Acids Res* **39**, D456-64 (2011).