Open-Source Image Analysis Connectivity for OMERO

David R Stirling[1], Erin Diel[1], Emil Rozbicki[1], Jason Swedlow[1,2]

[1]Glencoe Software Inc., Seattle, WA, USA
[2]The Open Microscopy Environment, University of Dundee, Dundee, UK

Background

- Scientists routinely capture **large, multi-dimensional datasets** containing millions of images.
- **OMERO** is an image data management platform developed by the Open Microscopy Environment (OME), specifically designed for handling life sciences data, including experimental metadata and analytics.
- **Python** is a popular programming language for data manipulation and analysis due to its extensive catalogue of third-party libraries tailored to the scientific community.
- We have been working to expand **interoperability** between OMERO servers and open-source image analysis tools.

OMERO Plus

Enterprise image database for scientific images and associated metadata. Supports more than 160 bio-image formats and together with OME-NGFF provides the first truly cloud native image data management solution.

OMERO.tables

PyTables-based system for storing/retrieving tabular data on OMERO servers, linked to relevant source image data.

CellProfiler

Open-source image analysis software maintained by the Broad Institute [2]. Uses modular pipelines to analyse image data, including high content screening datasets.

OMERO-CellProfiler Connector

Proprietary Glencoe Software tool for execution of CellProfiler pipelines remotely via OMERO clients. Supports various HPC systems including: SGE, PBS, LSF and SLURM and cloud deployment via AWS Batch.

Aim

Bridge the gap between OMERO and data science with a seamless connection to Python data science tools.

omero2pandas

omero2pandas is an open-source Python library designed to streamline data retrieval and storage by converting OMERO.tables to/from Pandas DataFrames.

Key features:

- Load OMERO.tables to DataFrame remotely

```
df = omero2pandas.read_table(file_id=402)
```

- Download the table for local use

```
omero2pandas.download_table(
    "/path/to/output.csv", file_id=2, chunk_size=1000)
```

- Upload a results table to OMERO

```
ann_id = omero2pandas.upload_table(
    my_data_frame, "Table Name", 142, "Image")
```

- Retrieve a list of columns from a remote table.

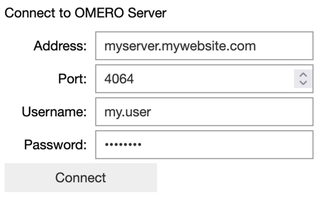
```
columns = omero2pandas.get_table_columns(
    annotation_id=142)
```

- Read specific rows and/or columns

```
my_dataframe = omero2pandas.read_table(
    file_id=10, column_names=['object', 'intensity'],
    rows=range(0, 100, 10))
```

- Server connection management, featuring automatic Jupyter environment detection and login widget

```
connector = omero2pandas.connect_to_omero(server="myserver.mywebsite.com")
```



- Connector also supports auth tokens generated with omero-user-token package.

CellProfiler plugins

For the upcoming release of CellProfiler 5, we have rebuilt OMERO functionality as a plugin with multiple new components:

OMEROReader – CellProfiler 5 will support modular image reader plugins, therefore reading of OMERO data has been developed as a standalone plugin which provides comprehensive access to data on OMERO.

SaveImagesToOMERO – A module plugin which allows images generated by CellProfiler to be uploaded directly onto OMERO.

ExportToOMEROTable – A module plugin to upload tabular data directly onto OMERO.

We have also included an OMERO browser. This can be used to directly add images to pipelines.

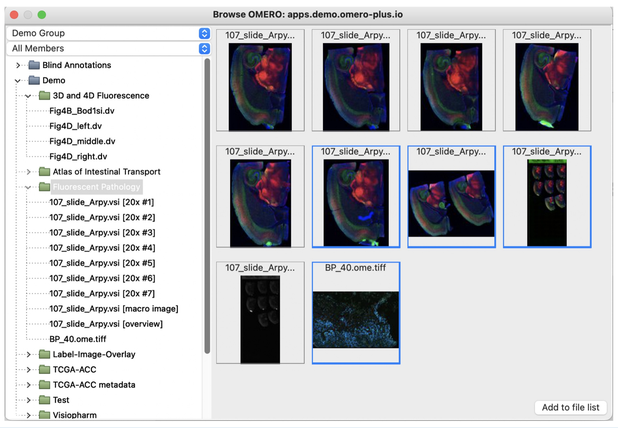


Figure 1: The new OMERO browser accessible from within CellProfiler (included with the OMERO plugins). Users can browse, select and add data to CellProfiler projects.

omero-user-token

Python package that creates long running user tokens for use with the OMERO API under non-interactive, headless conditions.

Create token:

```
omero_user_token set -s where.is.omero -u your.username --time_to_idle 0
```

Use the token:

```
omero_user_token get
```

```
from omero_user_token import getter
token = getter()
```

Use Case: Data

Supervised learning for HCS assays

Dataset: Human MCF7 and A549 cells cytoplasm–nucleus translocation.

The experiment demonstrates nuclear translocation of the transcription factor NF-κB in MCF7 and A549 cells in response to TNF-α concentration.

Goal:

Train a machine learning model to automatically classify cells based on NF-κB translocation.

Data:

- BBBC014: Human MCF7 (human breast adenocarcinoma) and A549 (human alveolar basal epithelial) cell lines
- 96-well plate, 10x objective magnification, a nuclear counterstain (DAPI) and NF-κB stain (FITC)
- Original images: 1360 x 1024 pixels, 8-bit BMP format were converted to OME-NGFF format before uploading to OMERO Plus

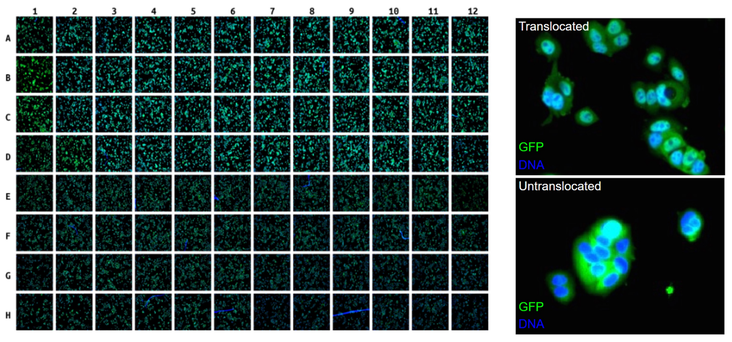


Figure 1: Images from the BBBC014 dataset, in 96-well plate format. https://bbbc.broadinstitute.org/BBBC014 [1]

Use Case: Analysis

Supervised learning for HCS assays

Images were segmented using Glencoe's OMERO-CellProfiler Connector. The process identified 70k cells. More than 150 shape and intensity-based features were measured to describe each cell.

Supervised classification pipeline [4]:

- Load segmentation results from OMERO.tables to Pandas DataFrame using the *omero2pandas* package
- Pre-process the data by replacing missing values and normalizing all features
- Select relevant features using a LinearSVC
- Build a cell classifier based on the positive and negative control wells
- Classify cells as translocated or untranslocated and count each type per concentration level.
- Save the result to OMERO.tables for visualization with OMERO Parade and further downstream workflows

Results:

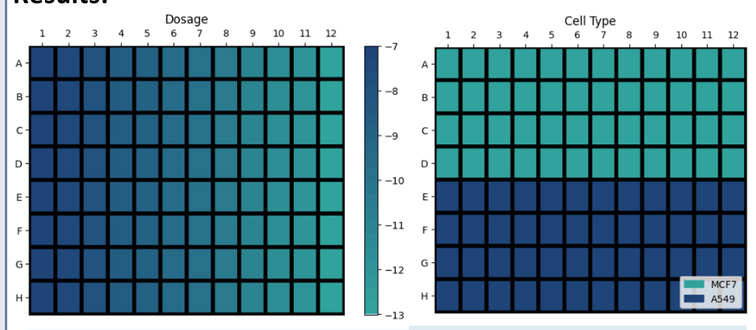


Figure 3: Experimental layout. Left: Concentration of TNFα (log g/ml). DMSO controls are represented as -13 units. Right: seeding of the MCF7 and A549 cells in the plate.

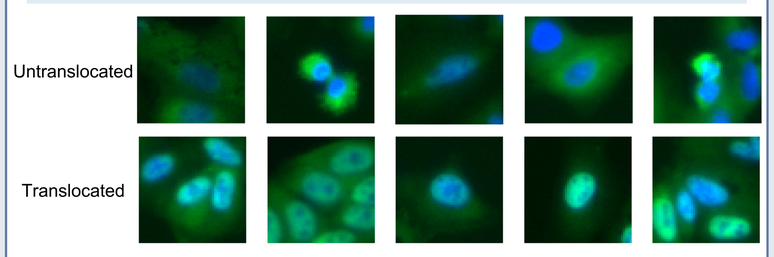


Figure 4: Images of representative A549 cells sampled randomly from the translocated/untranslocated populations as determined by the SVM classifier. Blue: DAPI, Green: FITC.

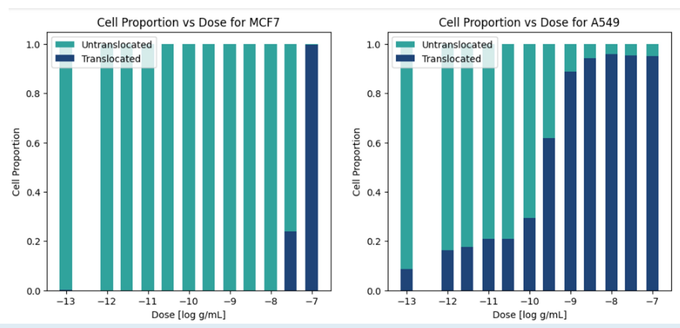


Figure 5: Analysis results showing the proportion of NF-κB translocated cells at each TNF-α concentration. Left: MCF7 cell line. Right: A549 cell line.

- Our classifier was able to effectively distinguish between translocated and untranslocated cells.
- The NF-κB translocation response to TNF-α was dose-dependent, but more pronounced in the A549 cells.

Conclusions

- We have developed open-source resources to enhance connectivity between OMERO and other tools.
- The CellProfiler 5 OMERO plugins will simplify accessing remote data from this software.
- Omero2pandas provides a convenient interface for exchanging tabular data with OMERO.

Code accessibility

https://github.com/glencoesoftware/omero2pandas
https://github.com/glencoesoftware/omero-user-token

References

1. We used image set BBBC014v1 provided by Ilya Ravkin, available from the Broad Bioimage Benchmark Collection [Ljosa et al., Nature Methods, 2012]
2. Stirling, D.R., Swain-Bowden, M.J., Lucas, A.M. *et al.* CellProfiler 4: improvements in speed, utility and usability. *BMC Bioinformatics* **22**, 433 (2021)
3. Weisbart, E., Tromans-Coia, C., Diaz-Rohrer, *et al.* CellProfiler plugins – An easy image analysis platform integration for containers and Python tools. *Journal of Microscopy*, **00**, 1–8. (2023)
4. www.github.com/glencoesoftware/webinar-notebooks