Utilising OME-NGFF to achieve scalable cloud-based analysis with CellProfiler

David R Stirling[1], Mina Gheiratmand[1], Chris MacLeod[1], Emil Rozbicki[1], Jason Swedlow[1,2]

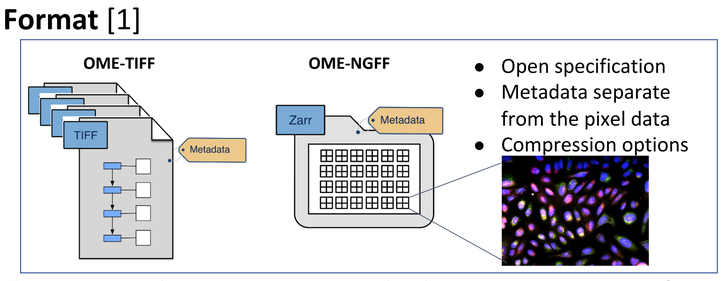
[1]Glencoe Software Inc., Seattle, WA, USA
[2]The Open Microscopy Environment, University of Dundee, Dundee, UK

Background

- Scientists routinely capture **large, multi-dimensional datasets** containing millions of images.
- Competing and frequently proprietary **storage formats** have been developed to handle this data.
- Existing formats are typically designed for use with **local storage**.
- Datasets can now exceed the capacity of local storage.
- Elastic **cloud-based storage** can handle large bioimage datasets.
- **Individual files** per image or per tile are convenient for accessing single frames but lack contextual metadata.

OME-NGFF

Open Microscopy Environment's Next-Generation File Format [1]



Open, vendor agnostic and domain agnostic format providing chunked, compressed, multi-dimensional data storage layout. Suitable for the local, network or cloud-based storage including **object storage** (Amazon S3, Azure Blob, etc.).

CellProfiler

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

OMERO Plus

Enterprise image database for scientific images and associated metadata. Supports more than 150 bio-image formats and together with OME-NGFF provides first truly cloud native image data management solution.

OMERO-CellProfiler Connector

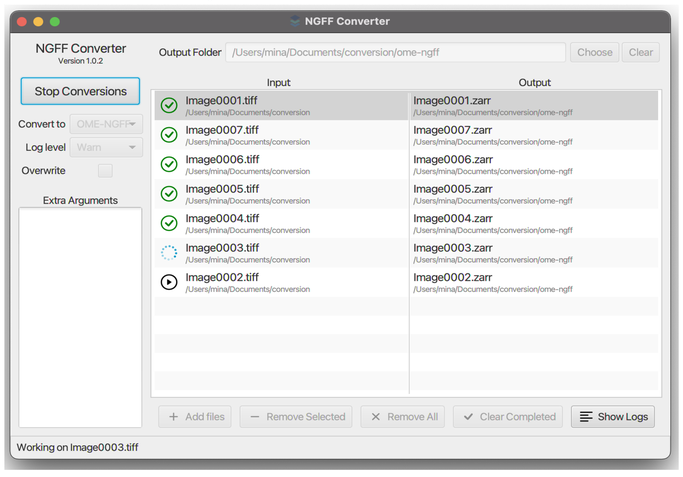
Proprietary Glencoe Software tool for execution of CellProfiler pipelines remotely via OMERO clients. Supports various HPC systems including: SGE, PBS, LSF and SLURM and cloud deployment via AWS Batch.

Aim

Evaluate suitability and performance of the OME-NGFF format for the HCS local and cloud-based image processing and analysis.

Methods

TIFF files from the public dataset **BBBC022** were converted to OME-NGFF format with **NGFF-Converter**. NGFF-Converter is Glencoe's open-source interface for the bioformats2raw and raw2ometiff packages which can generate OME-NGFF datasets from most bioimage formats.



https://glencoesoftware.com/products/ngff-converter

The **CellProfiler OME-NGFF reader** was implemented on a fork of the main repository (Broad Institute). The reader was developed for CellProfiler 4.2.1 (Python 3.8), using the zarr and fsspec libraries. Prebuilt binaries with this reader are available at:
https://github.com/glencoesoftware/CellProfiler/releases

Testing

- Performed local and cloud-based CellProfiler runs on **3456 image sets from BBBC022.**
- **Same data** in TIFF and OME-NGFF format stored on a local disk and in AWS S3 object storage.
- Reader performance evaluated based on LoadData module execution time.
- AWS Batch execution orchestrated by OMERO-CellProfiler Connector.

Results

Local CellProfiler execution

Setup: Workstation outside of AWS infrastructure with 16 CPUs and 32GB RAM. Copy of the test data stored on the local hard drive and AWS S3 storage.

CPU Time: A measure of computational workload (excludes I/O)
Wall Time: Total execution time as seen by the user

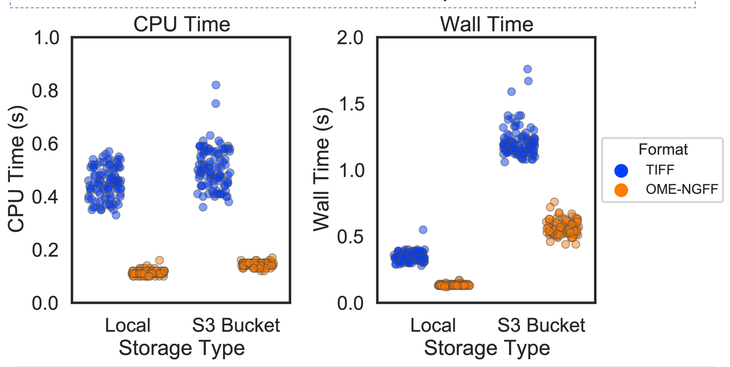
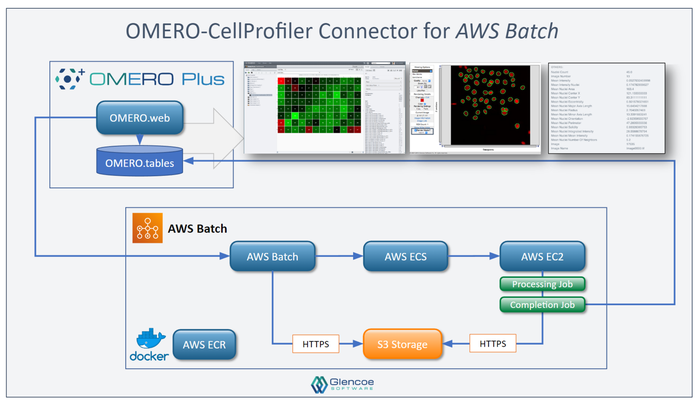


Figure 1: Comparison of LoadData module execution time when using Bio-Formats (TIFF) vs the OME-NGFF reader. Displayed data represents a random sampling of 100 image sets from the analysis run.

- Loading data from OME-NGFF provided a substantial performance advantage in all conditions (**Figure 1**).
- CPU time required to load the data was similar between local and S3 storage.

Cloud-based CellProfiler execution

Setup: AWS Batch infrastructure with the maximum of 256 vCPUs was used to analyse data stored in AWS object storage (S3 storage). OMERO-CellProfiler Connector was used to orchestrate the analysis with variable batch sizes (image sets per job).



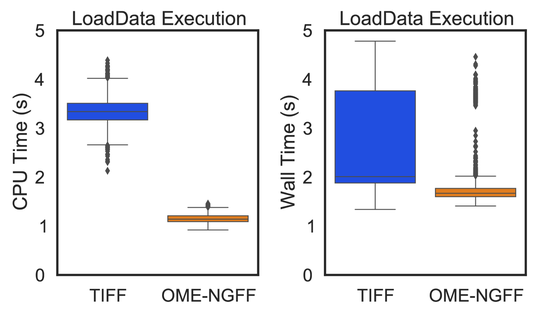


Figure 2: LoadData execution times for TIFF and OME-NGFF data when analysed in a cluster computing environment. Each node was assigned a single image set to analyse for each CellProfiler job. Results represent all 3456 image sets (1 CellProfiler job per image set).

- Similar performance advantage using the OME-NGFF format (**Figure 2**) on a computing cluster.
- Wall Time required for each image set was more variable than when running on a local machine.

Why is loading S3 bucket data slower on a cluster than on a local machine??

- Cluster workflows treat each image set as an **independent** CellProfiler run.
- CellProfiler process is restarted after each image set.
- This approach provides **resilience** against errors encountered during analysis.
- Restarting the process requires that image readers are **re-initialized** for each job.

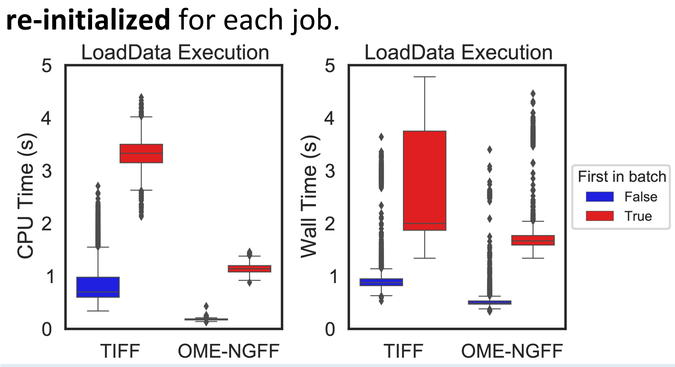


Figure 3: Execution times for TIFF and OME-NGFF data in a cluster computing environment. Analysis runs were performed with different batch sizes. Results are broken down into image sets which were the first to be analysed within a given execution vs subsequent image sets.

- The first image set in a batch was prone to slower performance (**Figure 3**).

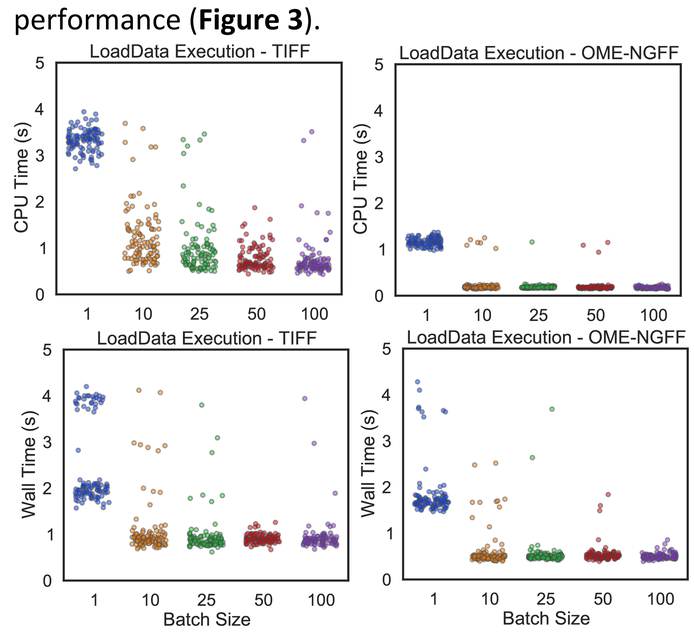


Figure 4: Comparison of LoadData execution time on AWS when using Bio-Formats (TIFF) vs the OME-NGFF reader, broken down by batch size. Data represents a random sampling of 100 image sets from the analysis run per batch size.

- Increasing batch size improved performance in both CPU and Wall time (**Figure 4**).
- OMERO-CellProfiler Connector can **automatically adjust batch size** to suit each dataset.
- OME-NGFF format maintained a **performance advantage** across all conditions.

Conclusions

- OME-NGFF can improve analysis performance when using CellProfiler both at the local and cluster level.
- Batch size plays an important role in determining the overall efficiency of data I/O operations within a pipeline.
  - Implications for configuration of existing cluster workflows such as Distributed-CellProfiler.
  - Can be resolved using automated batching such as in OMERO-CellProfiler Connector.
- CellProfiler 5 will introduce support for modular image readers.
- **Future work**: integrate the OME-NGFF reader into the main CellProfiler repository.

References

1 - Moore, J., Allan, C., Besson, S. *et al.* OME-NGFF: a next-generation file format for expanding bioimaging data-access strategies. *Nat Methods* **18**, 1496–1498 (2021).
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)