

ez-zarr: A Python package for easy access and visualisation of OME-Zarr filesets

Silvia Barbiero  ^{1,2*}, Charlotte Soneson  ^{1,3*}, Prisca Liberali  ^{1,2,4}, and Michael B. Stadler  ^{1,2,3*¶}

1 Friedrich Miescher Institute for Biomedical Research, Basel, Switzerland  2 University of Basel, Basel, Switzerland  3 SIB Swiss Institute of Bioinformatics, Basel, Switzerland  4 ETH Zürich, Department for Biosystems Science and Engineering (D-BSSE), Basel, Switzerland  ¶ Corresponding author * These authors contributed equally.

DOI: [10.21105/joss.07882](https://doi.org/10.21105/joss.07882)

Software

- [Review ↗](#)
- [Repository ↗](#)
- [Archive ↗](#)

Editor: Adam Tyson  

Reviewers:

- [@alessandrofelder](#)
- [@pr4depr](#)
- [@ianhi](#)

Submitted: 06 February 2025

Published: 17 May 2025

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

Summary

Imaging techniques are an important and growing source of scientific data, but the many different standards and file formats for representing and storing such data make streamlined analysis and data sharing cumbersome. To address this issue, the imaging community has recently developed an open and flexible file format specification (OME-NGFF) (Moore et al., 2021) and a concrete implementation using the Zarr file format (OME-Zarr) (Moore et al., 2023), which is increasingly adopted as the way to store and share multidimensional bioimaging data.

The aim of ez-zarr (pronounced “easy zarr”) is to provide high-level, read-only programmatic access to OME-Zarr files, abstracting away the complex, nested representation of these on disk. Internally, the OME-Zarr files are represented using a custom python class, which standardises simple operations such as metadata extraction, image subsetting and visualisation. OME-Zarr files with any number of dimensions are supported, and images with more than two spatial dimensions are automatically projected for plotting. In addition, the underlying image data, or subsets thereof, can be extracted as numpy arrays (Harris et al., 2020), which makes ez-zarr a valuable building block in more complex image analysis workflows that make use of tools that may not yet support OME-Zarrs.

ez-zarr also facilitates working with collections of OME-Zarr files, such as images from high-content screens where each microwell in a plate is represented by one OME-Zarr file and the OME-Zarr plate collection provides necessary metadata. Such plate-based high-content screening experiments are becoming increasingly popular in research, and entire analysis workflows, like Fractal ([BioVision Center, University of Zurich, 2025](#)), now streamline the generation and processing of OME-Zarr datasets.

Finally, while ez-zarr is implemented in python and installable using pip and conda, we also provide an R wrapper package called ezzarr that enables the use of the python functions from R (<https://github.com/fmicompbio/ezzarr>).

Statement of need

OME-Zarr is a relatively new file format and not yet supported by many image analysis tools. There are tools that allow accessing OME-Zarr files, amongst them interactive viewers, such as napari ([Sofroniew et al., 2024; Steffen & Luethi, 2025](#)) and MoBIE ([Pape et al., 2023](#)), and low level libraries to read and write OME-Zarrs, including ome-zarr ([The Open Microscopy Team, 2025](#)), bioio-ome-zarr ([The BioIO developers, 2025](#)), ngio ([Cerrone, 2025](#)), ngff-zarr ([McCormick, 2025](#)) and zarr ([Miles, 2025](#)).

While these tools allow interactive exploration of images in OME-Zarr format, or getting low-level access to the contained data, we were not aware of tools that provide higher-level access and scriptable visualisation capabilities. Some more specialised projects also cover such tasks, but focus on specific types of imaging data, for example spatial omics data in the `SpatialData` framework (Marconato et al., 2024). Furthermore, most available tools are implemented in python and do not have an equivalent interface in other languages like R. `ez-zarr` was designed to fill these gaps, building on the low-level file access functionality from the `zarr` library, and provide a minimal, abstract interface to OME-Zarr images or collections of such images from both python and R.

`ez-zarr` has been used productively by several research groups and in published research (Schwayer et al., 2025).

Examples

A typical use-case for `ez-zarr` is the scripted exploration of microscopy data at different resolutions, for example to perform quality control and to visualise the underlying imaging data for specific segmented objects. Figure 1 shows three example images, each created with a single `ez-zarr` function call.

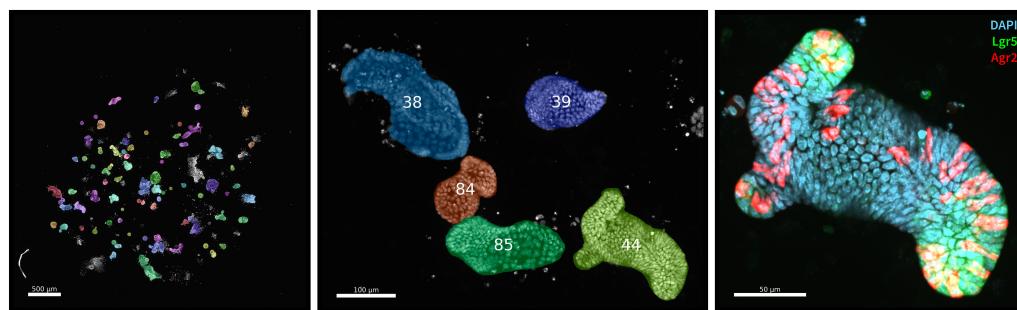


Figure 1: Three different plots with scale bars created from the same image: The full image with segmentation labels on the left, a zoomed-in version showing only a few objects in the middle, and a single object with different stainings shown as a false-colour image on the right.

For collections of images, for example obtained from high-content microscopy experiments, `ez-zarr` provides a special `ImageList` container that streamlines import and visualisation of entire microwell plates (Figure 2). The `ImageList` class also implements subsetting and parallelisation over its items.

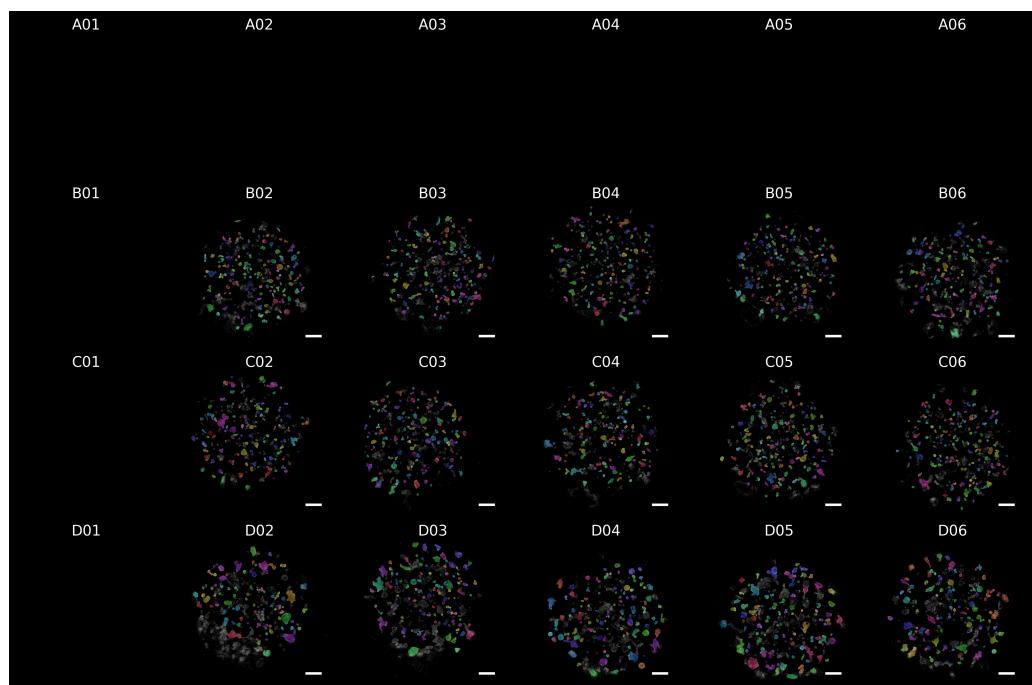


Figure 2: Example of a 24-well plate, in which only 15 of the wells were used in the experiment. The underlying data (a folder containing 15 OME-Zarrs) can be directly imported and visualised using ez-zarr. Here, we have additionally overlaid the intensity images with their corresponding segmentation labels, also stored in the OME-Zarr.

Acknowledgements

We would like to thank Joel Lüthi, Jan Eglinger, Tim-Oliver Buchholz and the members of the Liberali lab for fruitful discussions, testing and feedback on the software.

References

- BioVision Center, University of Zurich. (2025). *Fractal*. <https://fractal-analytics-platform.github.io/>.
- Cerrone, L. (2025). *ngio*. <https://pypi.org/project/ngio/>.
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- Marconato, L., Palla, G., Yamauchi, K. A., Virshup, I., Heidari, E., Treis, T., Vierdag, W.-M., Toth, M., Stockhaus, S., Shrestha, R. B., Rombaut, B., Pollaris, L., Lehner, L., Vöhringer, H., Kats, I., Saeys, Y., Saka, S. K., Huber, W., Gerstung, M., ... Stegle, O. (2024). SpatialData: An open and universal data framework for spatial omics. *Nature Methods*, 1–5. <https://doi.org/10.1038/s41592-024-02212-x>
- McCormick, M. (2025). *ngff-zarr*. <https://pypi.org/project/ngff-zarr/>.
- Miles, A. (2025). *zarr*. <https://pypi.org/project/zarr/>.
- Moore, J., Allan, C., Besson, S., Burel, J.-M., Diel, E., Gault, D., Kozlowski, K., Lindner,

- D., Linkert, M., Manz, T., Moore, W., Pape, C., Tischer, C., & Swedlow, J. R. (2021). OME-NGFF: A next-generation file format for expanding bioimaging data-access strategies. *Nature Methods*, 18(12), 1496–1498. <https://doi.org/10.1038/s41592-021-01326-w>
- Moore, J., Basurto-Lozada, D., Besson, S., Bogovic, J., Bragantini, J., Brown, E. M., Burel, J.-M., Casas Moreno, X., de Medeiros, G., Diel, E. E., Gault, D., Ghosh, S. S., Gold, I., Halchenko, Y. O., Hartley, M., Horsfall, D., Keller, M. S., Kittisopikul, M., Kovacs, G., ... Swedlow, J. R. (2023). OME-zarr: A cloud-optimized bioimaging file format with international community support. *Histochemistry and Cell Biology*, 160(3), 223–251. <https://doi.org/10.1007/s00418-023-02209-1>
- Pape, C., Meechan, K., Moreva, E., Schorb, M., Chiaruttini, N., Zinchenko, V., Martinez Ver-gara, H., Mizzon, G., Moore, J., Arendt, D., Kreshuk, A., Schwab, Y., & Tischer, C. (2023). MoBIE: A Fiji plugin for sharing and exploration of multi-modal cloud-hosted big image data. *Nature Methods*, 20(4), 475–476. <https://doi.org/10.1038/s41592-023-01776-4>
- Schwayer, C., Barbiero, S., Brückner, D. B., Baader, C., Repina, N. A., Diaz, O. E., Meylan, L. C., Kalck, V., Suppinger, S., Yang, Q., Schnabl, J., Kilik, U., Camp, J. G., Stockinger, B., Bühler, M., Stadler, M. B., Hannezo, E., & Liberali, P. (2025). Cell heterogeneity and fate bistability drive tissue patterning during intestinal regeneration. *bioRxiv*. <https://doi.org/10.1101/2025.01.14.632683>
- Sofroniew, N., Lambert, T., Bokota, G., Nunez-Iglesias, J., Sobolewski, P., Sweet, A., Gaifas, L., Evans, K., Burt, A., Doncila Pop, D., Yamauchi, K., Weber Mendonça, M., Buckley, G., Vierdag, W.-M., Royer, L., Can Solak, A., Harrington, K. I. S., Ahlers, J., Althviz Moré, D., ... Zhao, R. (2024). *napari: A multi-dimensional image viewer for python*. Zenodo. <https://doi.org/10.5281/zenodo.3555620>
- Steffen, F., & Luethi, J. (2025). *napari-ome-zarr-navigator*. <https://pypi.org/project/napari-ome-zarr-navigator/>.
- The BioIO developers. (2025). *bioio-ome-zarr*. <https://pypi.org/project/bioio-ome-zarr/>.
- The Open Microscopy Team. (2025). *ome-zarr*. <https://pypi.org/project/ome-zarr/>.