

OME-Arrow Unifying Images, Metadata, and Morphology in an Interoperable Data Model for High-Content Imaging

Dave Bunten¹, Jenna Tomkinson¹, Michael Lippincott¹, Cameron Mattson¹, Julia B. Curd¹, Gregory P. Way¹

¹Department of Biomedical Informatics, University of Colorado Anschutz Medical Campus



Anschutz

I. Why OME-Arrow?

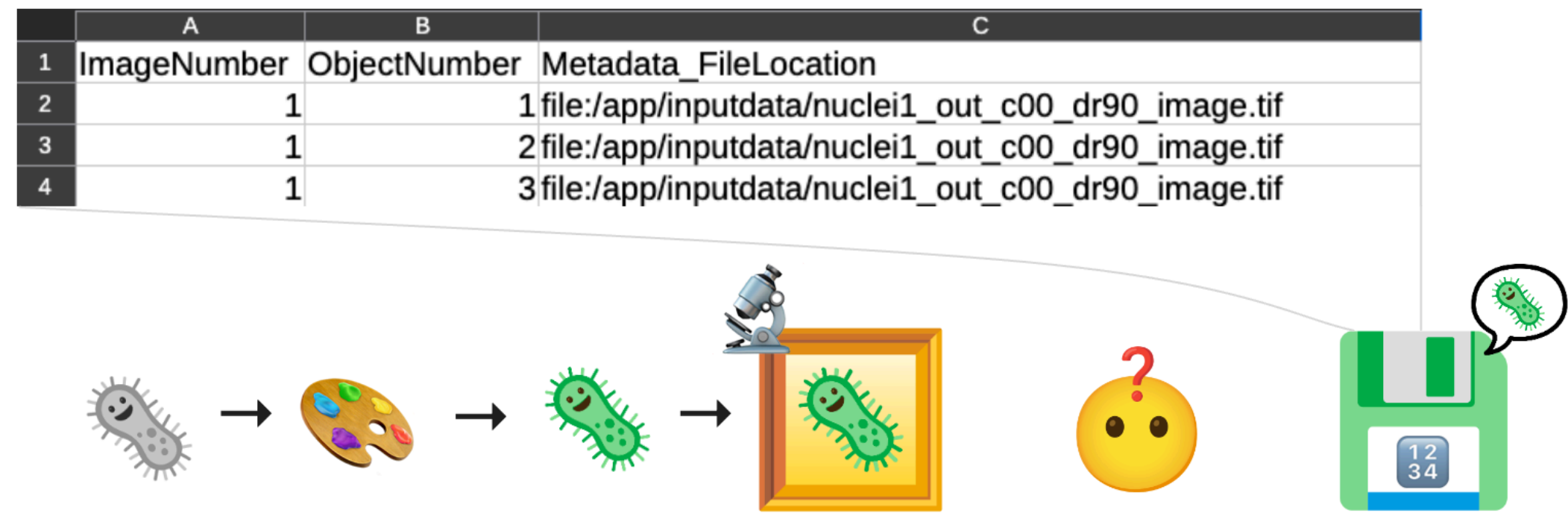


Figure 1: Modern bioimaging workflows depend on connecting images, metadata, and derived measurements. When these are split across disconnected files and systems, analysis is harder to join, reproduce, and scale. **OME-Arrow** provides a linked, queryable data model for these components in code-first and SQL-first analytical workflows.

II. An interoperable image data model

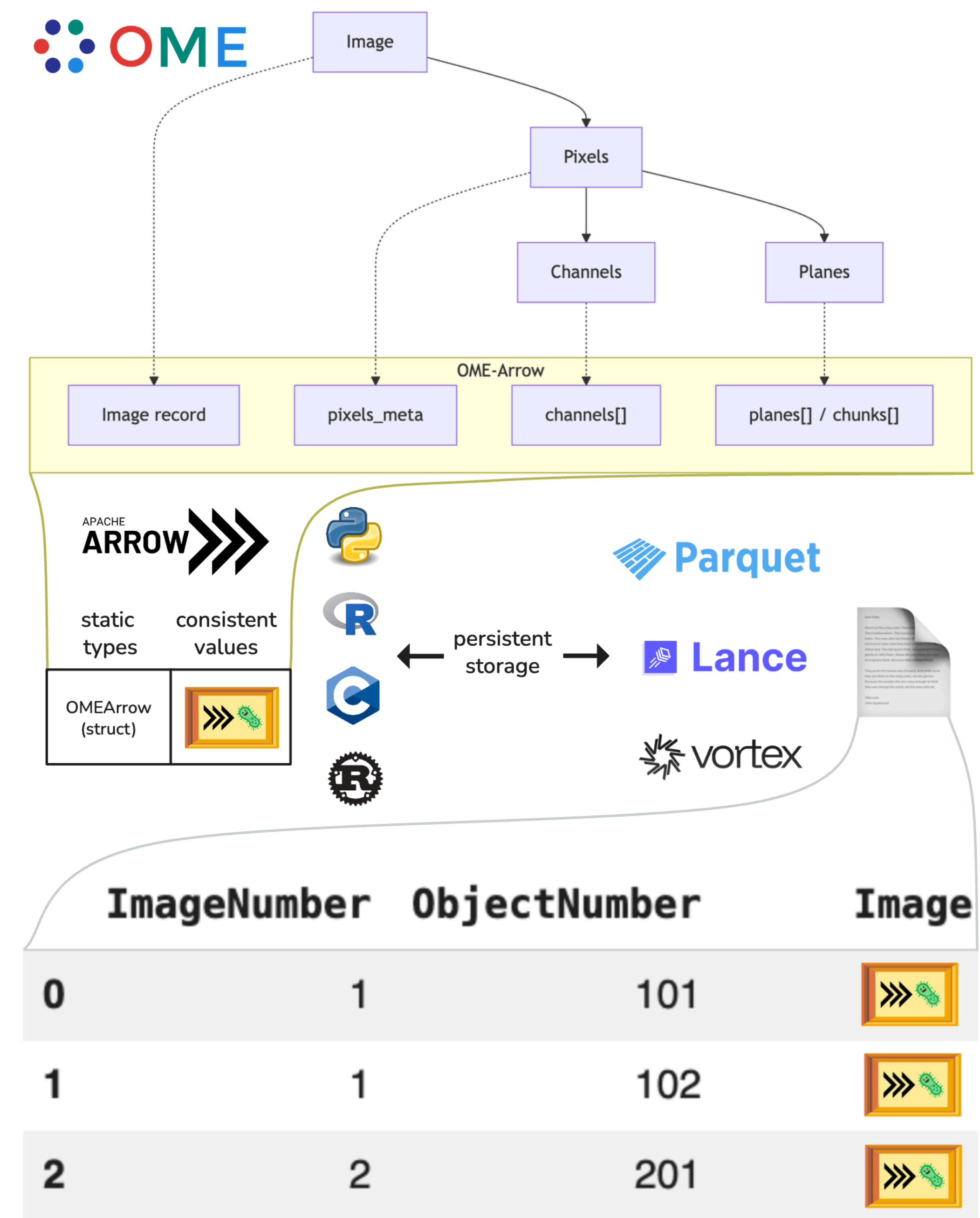


Figure 2: By representing image content as **Arrow-compatible** structures alongside metadata and features, **OME-Arrow** enables: **Explicit relationships** between pixels, metadata, and derived features, **Direct joins and filtering** in SQL and DataFrame workflows, **Cross-language interoperability** through Arrow-native representations, **Standards alignment** with the broader open bioimaging ecosystem

III. Benchmarks for OME-Arrow

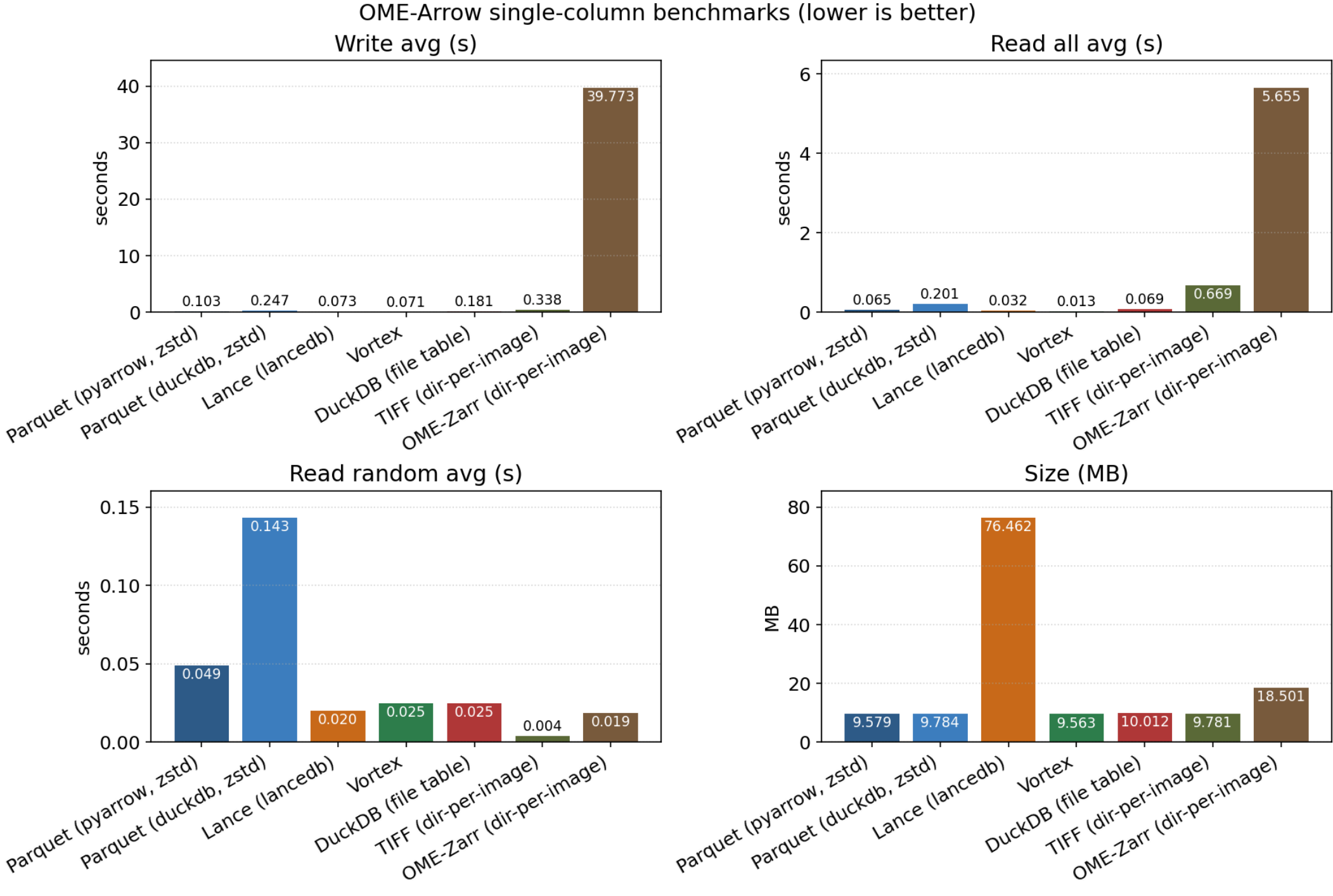


Figure 3: Current benchmark results suggest operation-dependent tradeoffs rather than one universal winner. In **ome-arrow-benchmarks**, OME-Zarr can perform strongly for sparse random image access patterns, while Arrow-table-native layouts provide practical advantages for table-centric join/filter workloads and broader analytical interoperability. Lance shows competitive random-access behavior in OME-Arrow-oriented tests, making it a strong candidate for large image-linked table repositories.

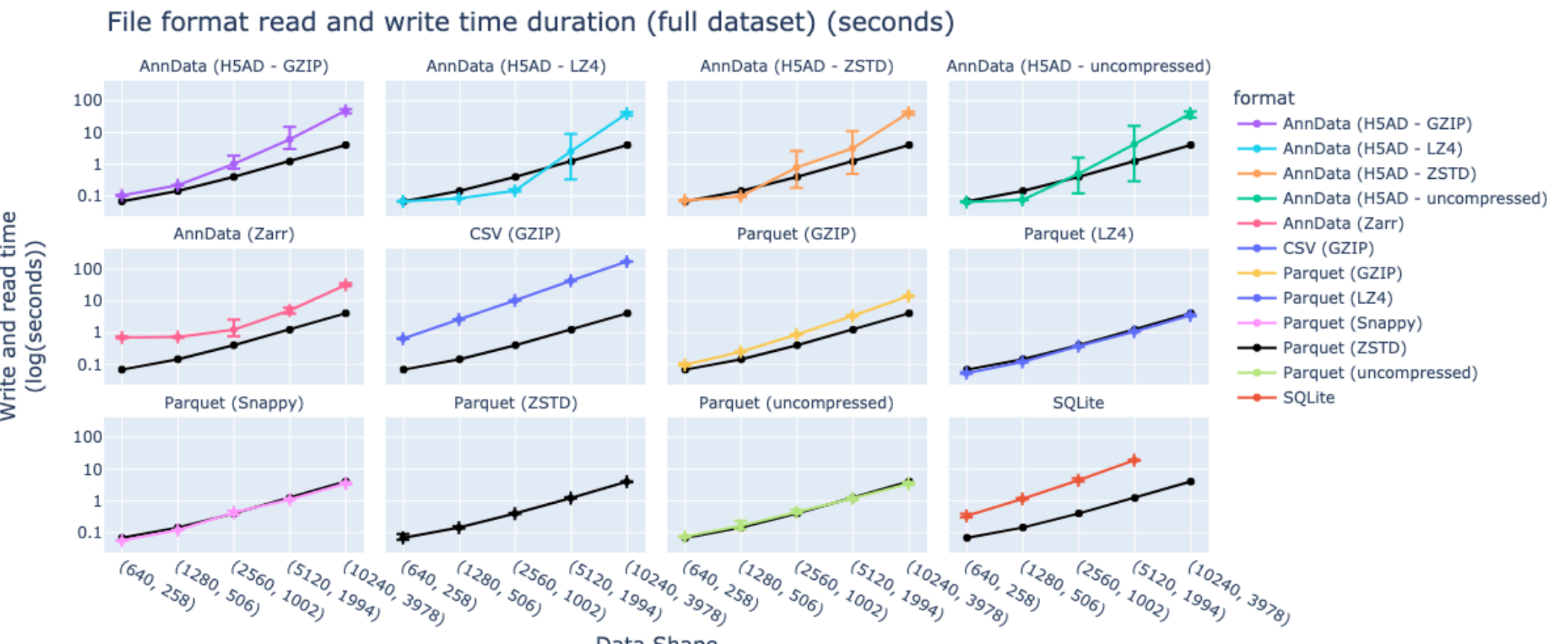


Figure 4: In **CytoTable-benchmarks**, Parquet and other Arrow-compatible tabular workflows generally scale better and run faster at larger data volumes than Zarr- and AnnData-oriented table paths, supporting the use of Arrow-native tables for large-scale profiling analytics.

IV. Quick technical demonstration

```
# install with `pip install ome_arrow`
from ome_arrow import OMEArrow

# Create an OME-Arrow object from an OME-Zarr
oa = OMEArrow("image.ome.zarr")

# Export to Parquet directly
oa.export(how="ome-parquet", out="image.parquet")

# Create and collect a "lazy" crop of an image
lazy = OMEArrow.scan("image.parquet").slice_lazy(0, 512, 0, 512).collect()

# visualize the image with pyvista for jupyter-friendly views
oa.view(how="pyvista")
```

V. Full Cytomining integration

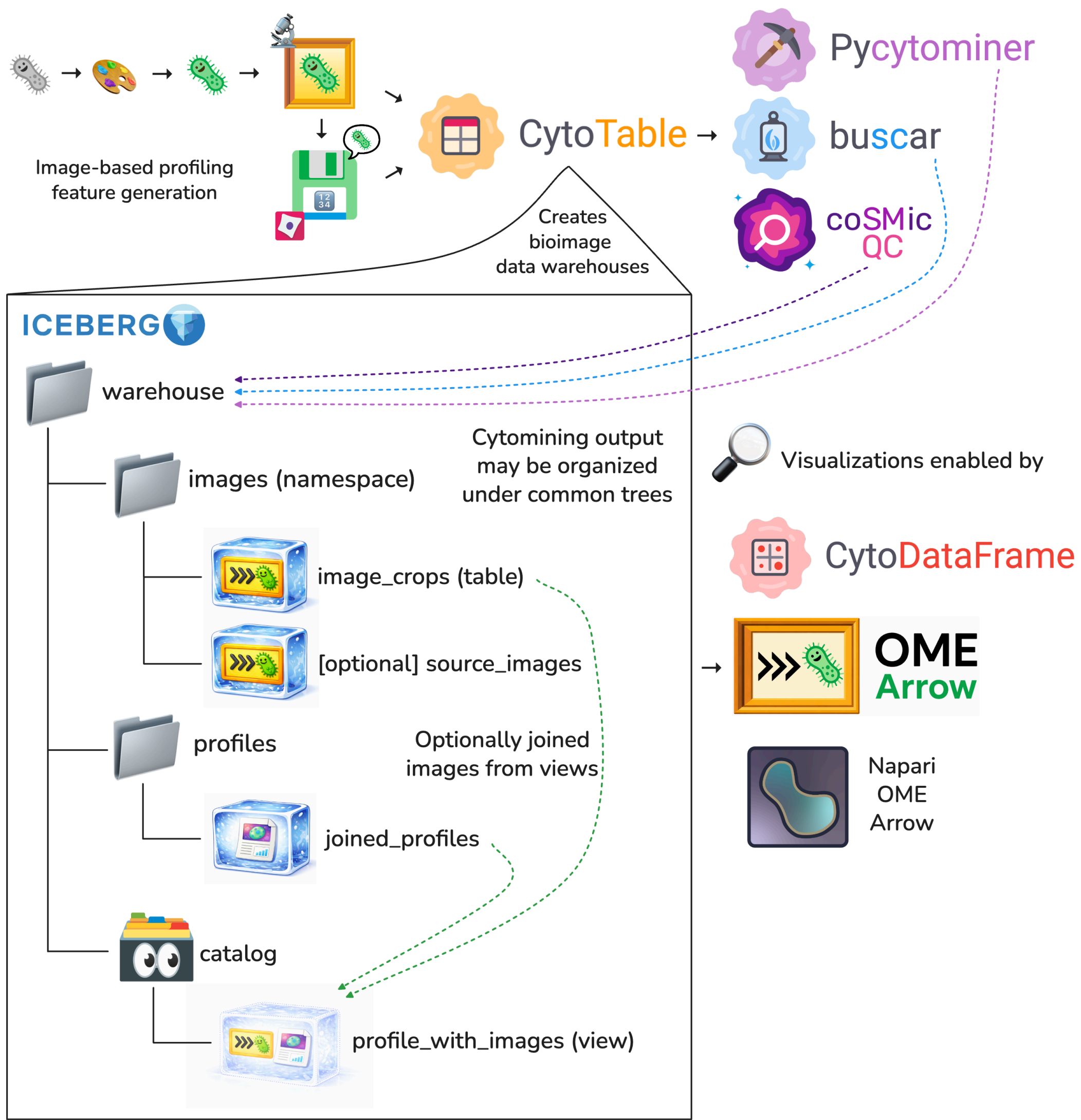


Figure 5: OME-Arrow complements Cytomining tools in a modular stack: OME-Arrow provides built-in visualization (**matplotlib**, **pyvista**) for direct inspection; **napari-ome-arrow** adds interactive viewing for OME-Arrow/OME-Parquet data; **CytoDataFrame** supports DataFrame-centered analysis of image-linked features and metadata; **coSMicQC** provides quality control with image context; **buscar** enables heterogeneity-aware single-cell compound ranking; and **pycytominer** supports profiling and normalization workflows.

VI. OME-Zarr and iceberg-bioimage

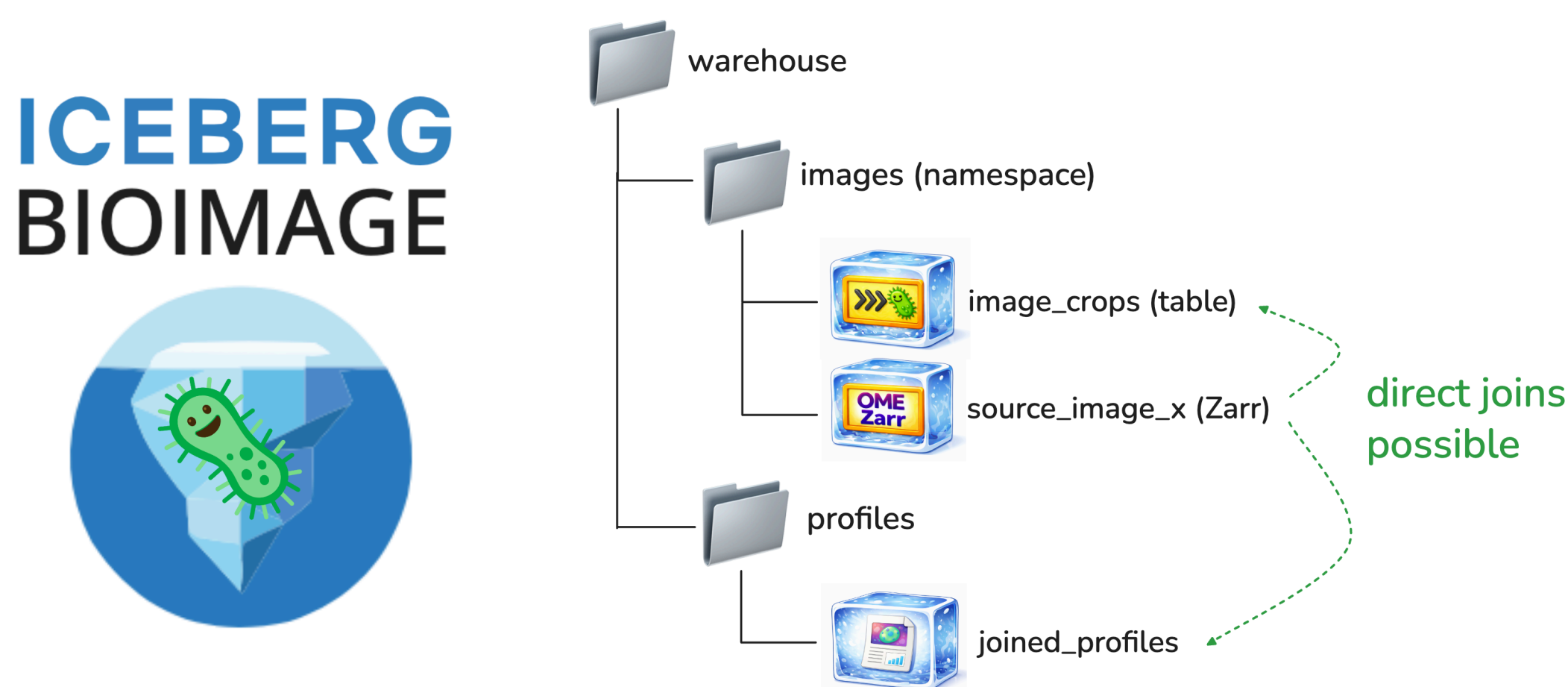


Figure 6: OME-Zarr is strong for cloud-native image storage and distribution. OME-Arrow is complementary (not a replacement) when images must be queried with tabular metadata and measurements. One integration pattern uses **iceberg-bioimage** as a warehouse/control-plane layer and **duckdb_zarr** for analytical access to OME-Zarr-backed data. Here, **OME-Zarr / OME-TIFF** remain exchange formats, while **OME-Arrow / Parquet / Arrow-native tables** support joins and analytics, with **Lance** as a random-access table option.

VII. ALSF pediatric cancer research

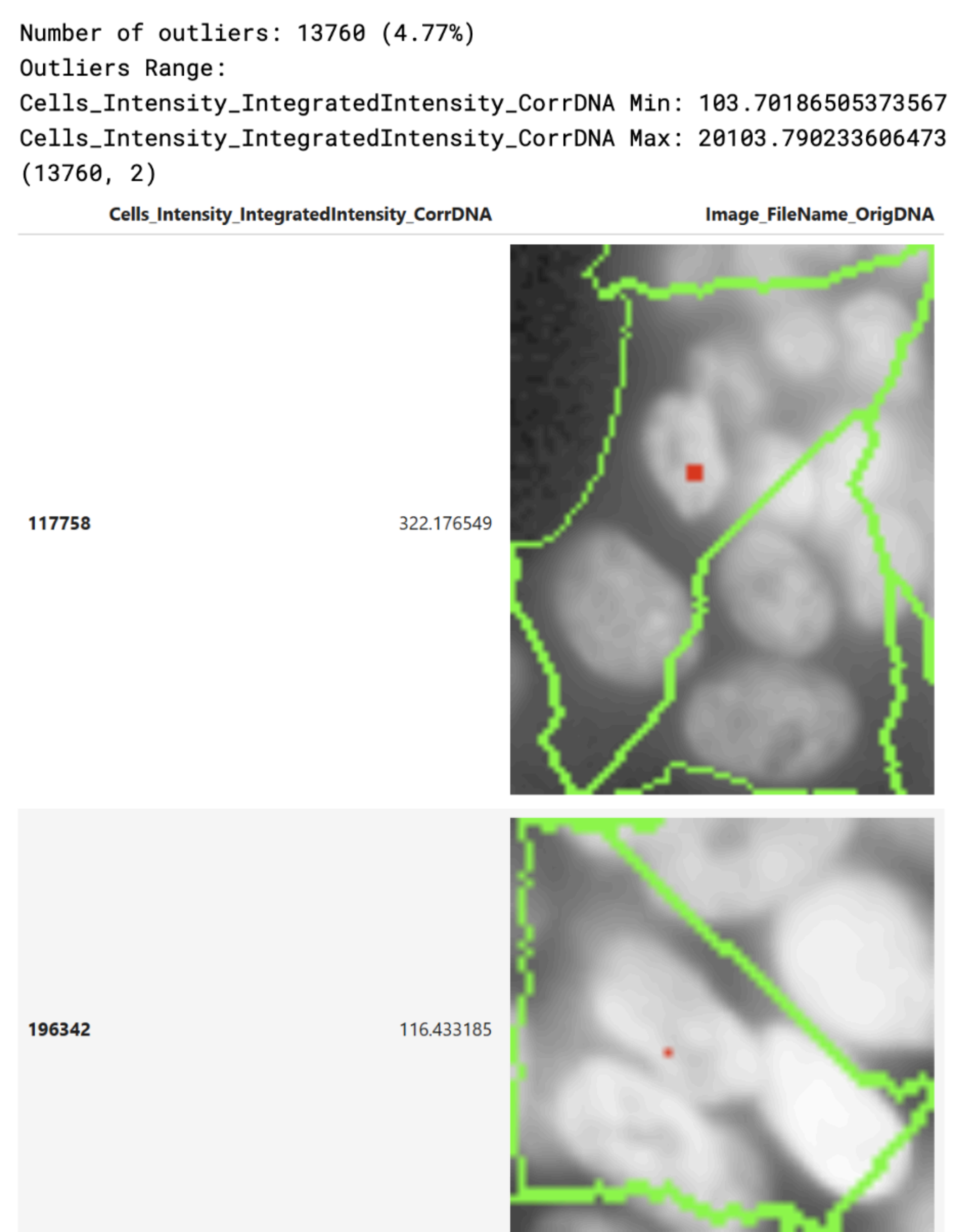


Figure 7: This work supports the **Pediatric Cancer Cell Morphology Atlas (PCCMA)** project, which will study pediatric cancer cell lines together with drug screens to accelerate discoveries that cure disease. The figure shows a **CytoDataFrame**-centered quality-control workflow using **coSMicQC** and Cytomining tooling on image-linked profiling data; in PCCMA analyses, this workflow helps identify and remove outliers for improved downstream analysis quality. We are using OME-Arrow-related work through the Cytomining ecosystem to help build the Atlas (for current implementation context, see [pediatric_cancer_atlas_profiling](#)).

VIII. Acknowledgements

We thank those who have inspired, contributed, or helped support OME-Arrow and the broader open bioimaging ecosystem:

- Open source science from the **Open Microscopy Environment**
- The **napari community** for open, collaborative development around interactive bioimage analysis
- The communities behind **Apache Arrow**, **Apache Iceberg**, and Cytomining ecosystem projects
- The **ALSF Childhood Cancer Data Lab** for pediatric cancer data science leadership and partnership
- Members of the **Way Lab** at the University of Colorado Anschutz Medical Campus
- **Department of Biomedical Informatics** within the School of Medicine at the University of Colorado Anschutz Medical Campus