CytoTable: High Performance and Scalable Single-cell Morphology Feature Engineering

Dave Bunten[1], Erik Serrano[1], Jenna Tomkinson[1], Michael J. Lippincott,[1], Faisal Alquaddoomi[1], Vince Rubinetti[1], Gregory P. Way[1]

[1] Department of Biomedical Informatics, University of Colorado Anschutz

Department of Biomedical Informatics
SCHOOL OF MEDICINE
UNIVERSITY OF COLORADO **ANSCHUTZ MEDICAL CAMPUS**

Introduction

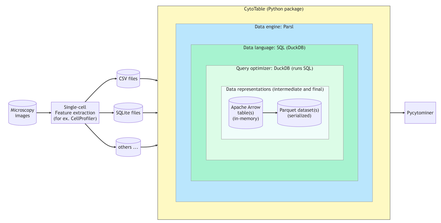

Figure 1. Diagram showing high-dimensional single-cell morphology data flow with relationship to CytoTable modular data stack components.

We are solving significant scalability and replicability challenges with high-dimensional single-cell morphology data (such as those extracted from CellProfiler[1]) by implementing novel and effective capabilities as a modular, portable, and cross-language single-cell data stack[2]: (a) language frontend: SQL (DuckDB[3]), (b) intermediate representation: Apache Arrow[4] and Apache Parquet[5], (c) query optimizer: DuckDB[3], (d) execution engine: Parsl[6] with Pythonic MapReduce design patterns[7], (e) execution runtime, Python package (PyPI, source)(Figure 1).

Morphology Feature Data Scalability

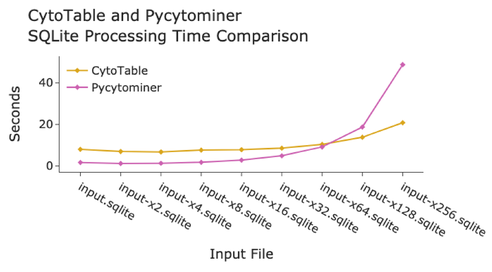

Figure 2. Plot showing processing time duration for CytoTable and Pycytominer for various datasets.

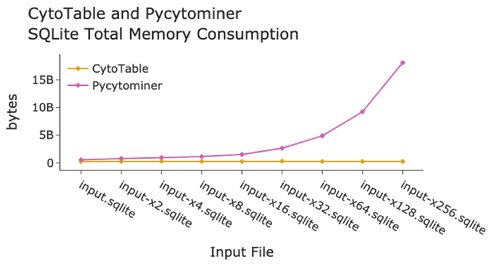

Figure 3. Plot showing total memory consumption for CytoTable and Pycytominer for various datasets.

CytoTable builds upon the shoulders of Pycytominer, helping to streamline the `SingleCells.merge_single_cells(...)` method. We decrease overall processsing completion time (Figure 2) and memory consumption (Figure 3) for large amounts of data by leveraging composable data stack elements.

Empowering the Cytomining Ecosystem

Orchestration: CytoSnake

Authors: Erik Serrano, Jenna Tomkinson, Roshan Kern, Vince Rubinetti, Dave Bunten, Gregory P. Way

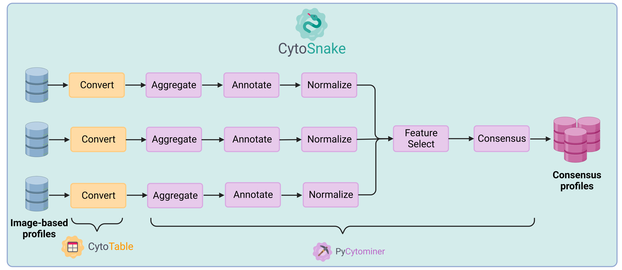

Figure 4. Diagram showing how CytoSnake orchestration applied in multiple pipelines.

CytoSnake is an innovative tool for orchestrating high-dimensional cell morphology data processing pipelines, including those which leverage CytoTable and other applied usecases.

Applied research: NF1 Schwann cell analysis

Authors: Jenna Tomkinson, Cameron Mattson, Erik Serrano, Gregory P. Way

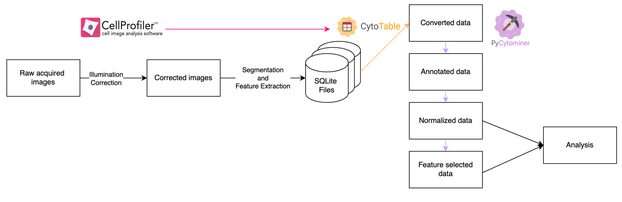

Figure 5. Diagram showing NF1 pipeline implementation details including CytoTable and Pycytominer.

Comprehensive dataset analyses for cell painting assays, enabling the further understanding of cellular morphology NF1 Schwann cells and rare disease treatment.

Applied research: Pyroptosis signature project

Authors: Michael J. Lippincott, Jenna Tomkinson, Gregory P. Way

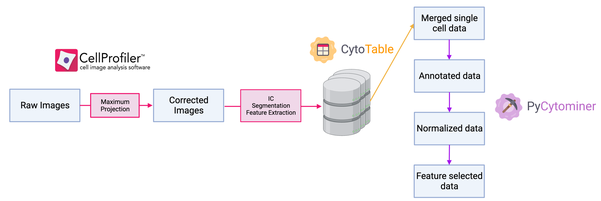

Figure 6. Diagram showing pyroptotic cell changes which are studied as part of the Pyroptosis signature project.

Identifying and characterizing pyroptosis signatures in cellular systems, aiding in the study of inflammatory cell death pathways as part of the Interstellar collaboration.

Applied research: CFReT project

Authors: Jenna Tomkinson, Erik Serrano, Gregory P. Way

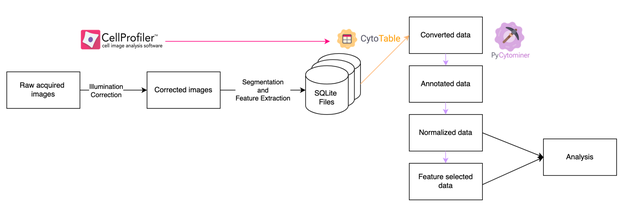

Figure 7. Diagram showing CFReT pipeline implementation details including CytoTable and Pycytominer.

Image-based analysis of cardiac fibroblast datasets to uncover proprietary drug impact on reversing fibrosis.

Using CytoTable

```
import cytotable
result_file = cytotable.convert(
    source_path="path/to/feature-data",
    dest_path="destination/path.parqut",
    dest_datatype="parquet",
    preset="cellprofiler_csv",
)
```

Figure 8. Code block showing Pythonic syntax for using CytoTable.

CytoTable includes a Pythonic API which can be customized as needed or leverage existing presets (Figure 8). See the CytoTable documentation for more detail: https://cytomining.github.io/CytoTable/

Shape the future with us!

Figure 9. Cytomining Ecosystem software logos.

The Cytomining Ecosystem cultivates groundbreaking science and research software engineering in the realm of high-dimensional single-cell science, guiding in a new era of bioinformatic innovation.

Interested in collaborating?
We welcome your input, contributions, and guidance!

Find us at https://github.com/cytomining.

References

1. Carpenter, A. E., Jones, T. R., Lamprecht, M. R., Clarke, C., Kang, I., Friman, O., Guertin, D. A., Chang, J., Lindquist, R. A., Moffat, J., Golland, P., & Sabatini, D. M. (2006). CellProfiler: Image analysis software for identifying and quantifying cell phenotypes. Genome Biology, 7(10), R100. https://doi.org/10.1186/gb-2006-7-10-r100
2. Pedreira, P., Erling, O., Karanasos, K., Schneider, S., McKinney, W., Valluri, S. R., Zait, M., & Nadeau, J. (2023). The Composable Data Management System Manifesto. Proceedings of the VLDB Endowment, 16(10), 2679–2685. https://doi.org/10.14778/3603581.3603604
3. Raasveldt, M., & Mühleisen, H. (2019). DuckDB: An Embeddable Analytical Database. Proceedings of the 2019 International Conference on Management of Data, 1981–1984. https://doi.org/10.1145/3299869.3320212
4. Apache Arrow. (n.d.). [Computer software]. Apache Software Foundation. https://arrow.apache.org/docs/
5. Apache Parquet. (n.d.). [Computer software]. Apache Software Foundation. https://parquet.apache.org/docs/
6. Babuji, Y., Woodard, A., Li, Z., Katz, D. S., Clifford, B., Kumar, R., Lacinski, L., Chard, R., Wozniak, J. M., Foster, I., Wilde, M., & Chard, K. (2019). Parsl: Pervasive Parallel Programming in Python. Proceedings of the 28th International Symposium on High-Performance Parallel and Distributed Computing, 25–36. https://doi.org/10.1145/3307681.3325400
7. Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. Communications of the ACM, 51(1), 107–113. https://doi.org/10.1145/1327452.1327492

Acknowledgements

Special thanks goes to the following for their help in contributing to CytoTable design and development or related work.

- Way Lab: Cameron Mattson
- Broad Institute: Shantanu Singh, Beth Cimini, Sam Chen
- Yale School of Medicine: Samir Amin