Cross-language Data Grammar for Single-cell Research Feature Engineering

Dave Bunten [1],
Faisal Alquaddoomi [1],
Gregory P. Way [1],

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

Introduction

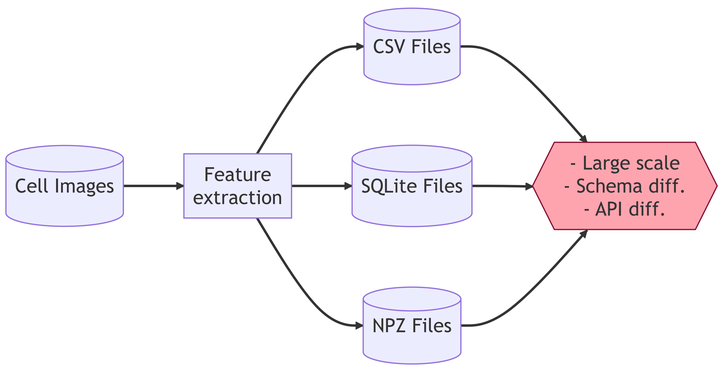


Figure 1. A diagram showing many different feature data and common challenges.

Research in the Way Lab involves intensive data engineering over high-dimensional single-cell morphology data from large-scale microscopy drug screening applications. Software development surrounding this work often entails scalability (larger than memory data handling) and understandability (syntax complexity and software sustainability) challenges.

Solution design

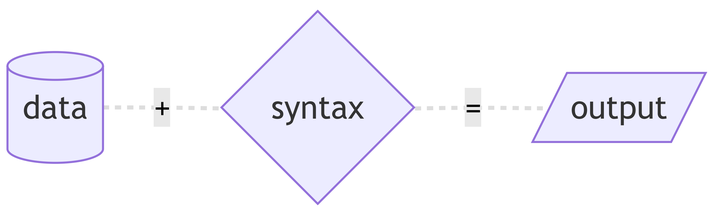
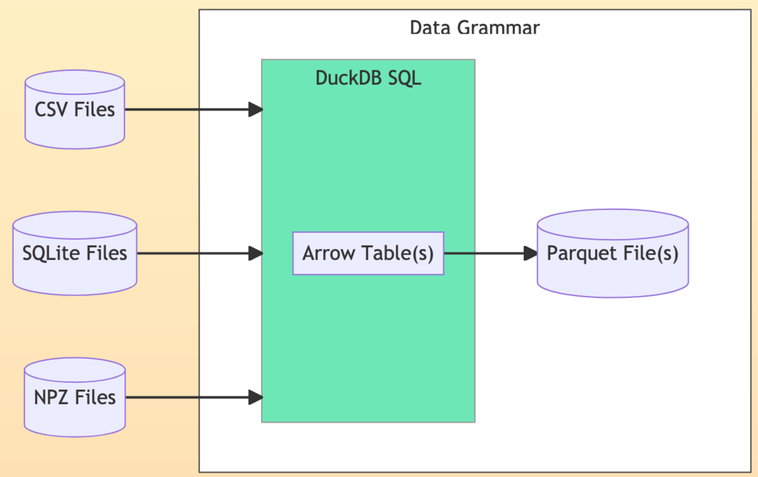


Figure 2. A diagram illustrating data grammar as an abstract linguistic algorithm.

To address these challenges we have developed a python package called CytoTable which implements "cross-language data grammar" capabilities (vocabulary + syntax = output) orchestrated with Parsl workflows. Our vision is for CytoTable to increase consistency and reliability of data and enable more scientists to quickly access single-cell insights from microscopy images.

CytoTable implements **data grammar** through **Apache Arrow**, **DuckDB SQL**, **Apache Parquet** for increased research velocity, cross-language integration, and understandability.



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

Data: Apache Arrow

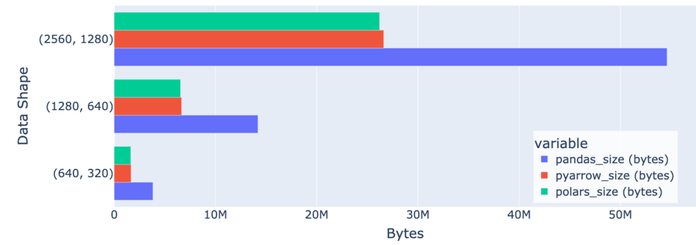


Figure 3. Chart showing relative memory size for data using various Python libraries.

Apache Arrow represents a new frontier for data implementation flexibility, enabling a unified, multi-language, zero-copy format for in-memory analysis. Arrow is like a high-performance Pandas dataframe which may be used across languages with fewer scalability challenges.

Syntax: DuckDB SQL

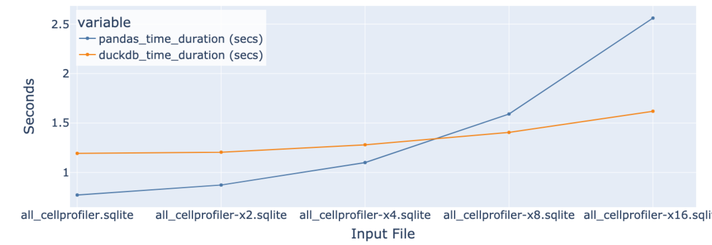


Figure 4. Chart showing read time durations of Pandas and DuckDB with SQLite databases of various sizes.

Structured Query Language (SQL) through DuckDB provides an Arrow-compatible embedded database system optimized for vectorized execution. DuckDB delivers in-memory capabilities through SQL, treating variable data as a loose collection of database tables without needing conversion.

Output: Apache Parquet

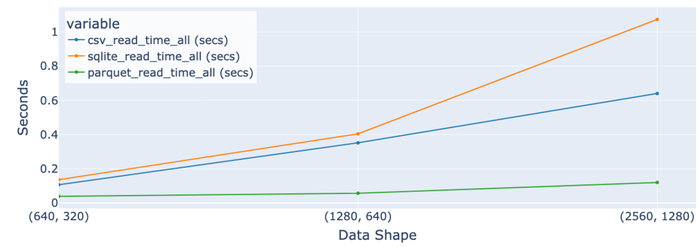


Figure 5. Chart showing read time durations for data from various file formats.

Work is saved in Apache Parquet files, which are compatible with Apache Arrow, and designed for storage and retrieval efficiency. Parquet is a columnar data format which may be partitioned across one or many files.

Acknowledgements

Special thanks goes to the following for their help in contributing to CytoTable, this poster, or related work.

- The Way Lab (https://www.waysciencelab.com/)
- The Broad Institute (https://www.broadinstitute.org/)
- Vince Rubinetti (https://cu-dbmi.github.io/set-website/members/vincent-rubinetti.html)