

# Cross-language Data Grammar for Single-cell Research Software Engineering

Dave Bunten<sup>1</sup>,   
Gregory P. Way<sup>1</sup>, 

<sup>1</sup> Department of Biomedical Informatics, University of Colorado Anschutz

## Introduction

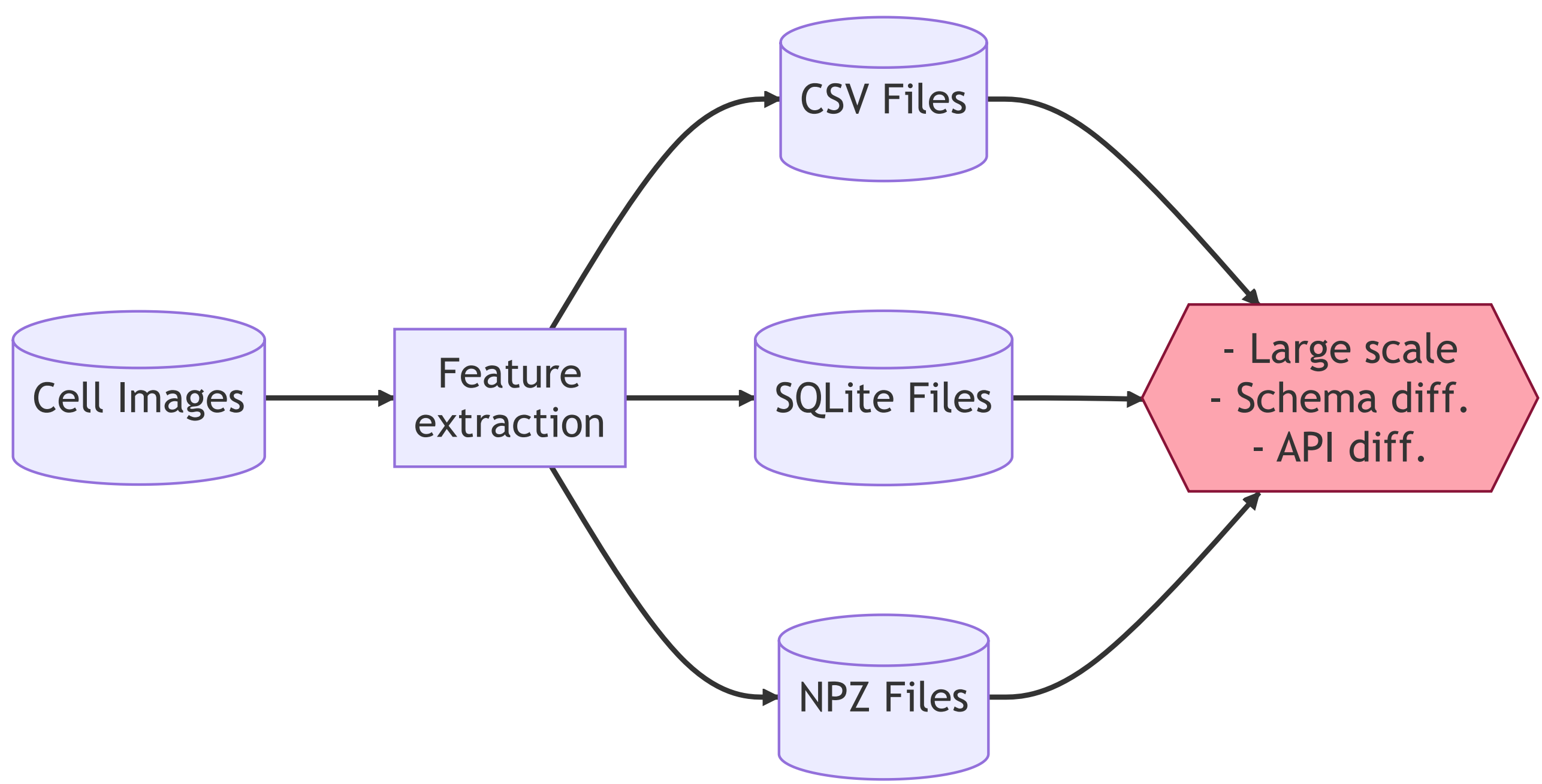


Figure 1. A diagram showing many different feature data and common challenges.

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

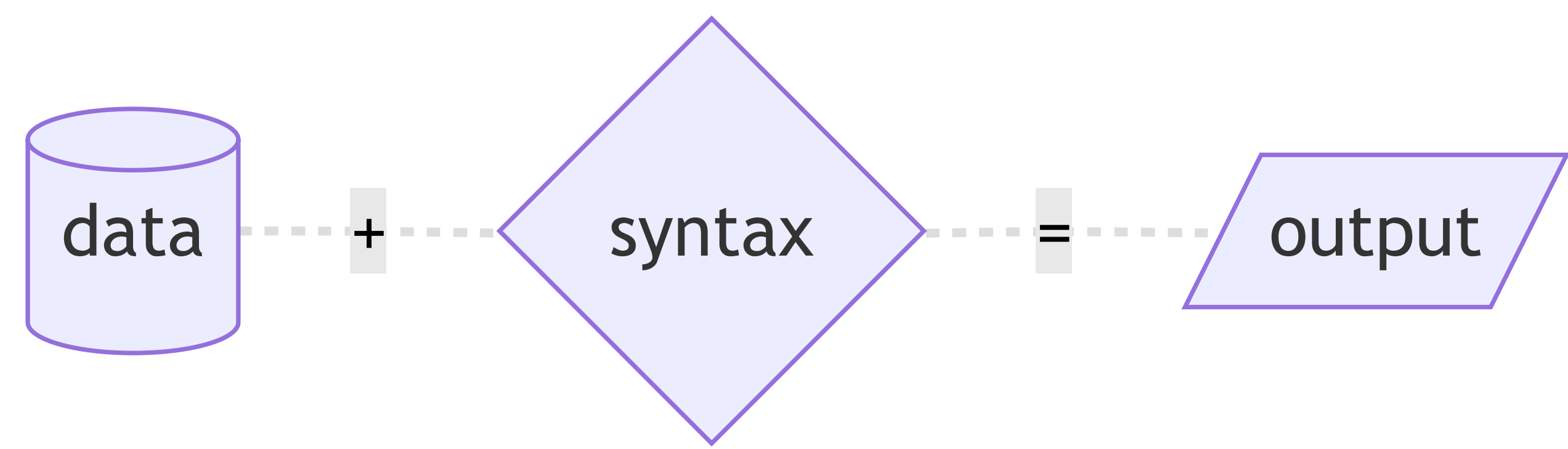


Figure 2. A diagram illustrating data grammar as an abstract linguistic algorithm.

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.

# Impelmenting data grammar through SQL, Arrow, Parquet, ...

## Data: Apache Arrow

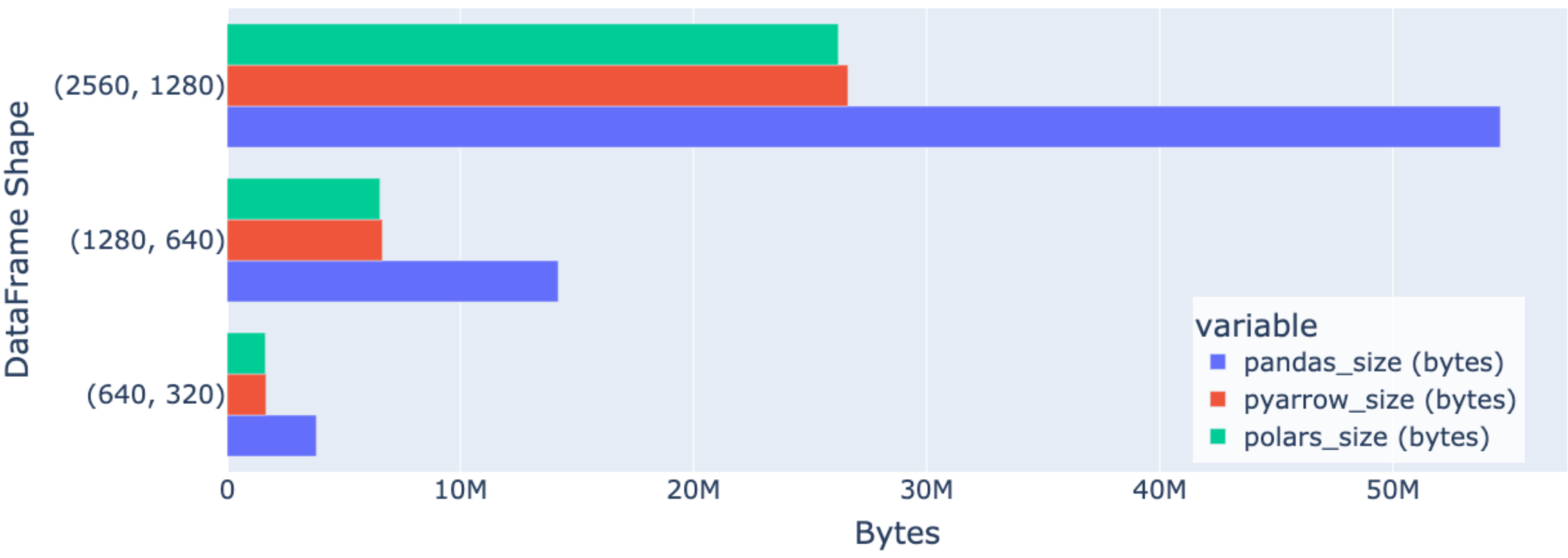


Figure 3. Chart showing relative memory size for data using various Python libraries.

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

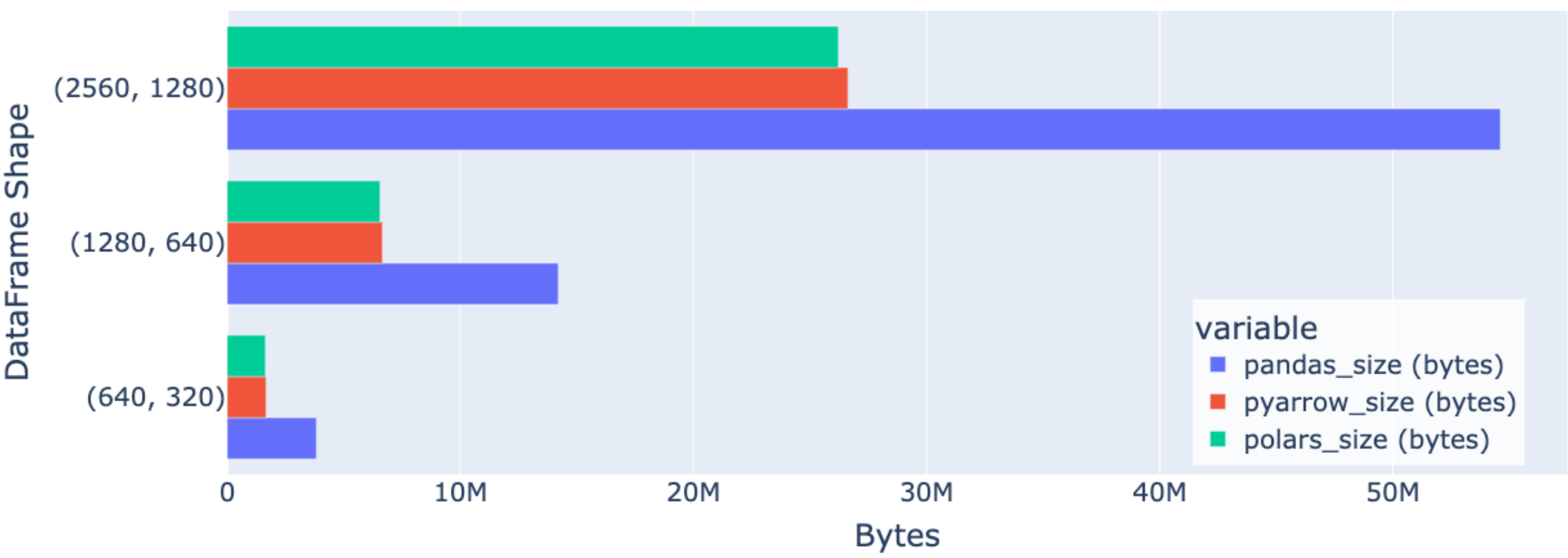


Figure 4.

Figure 4.

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

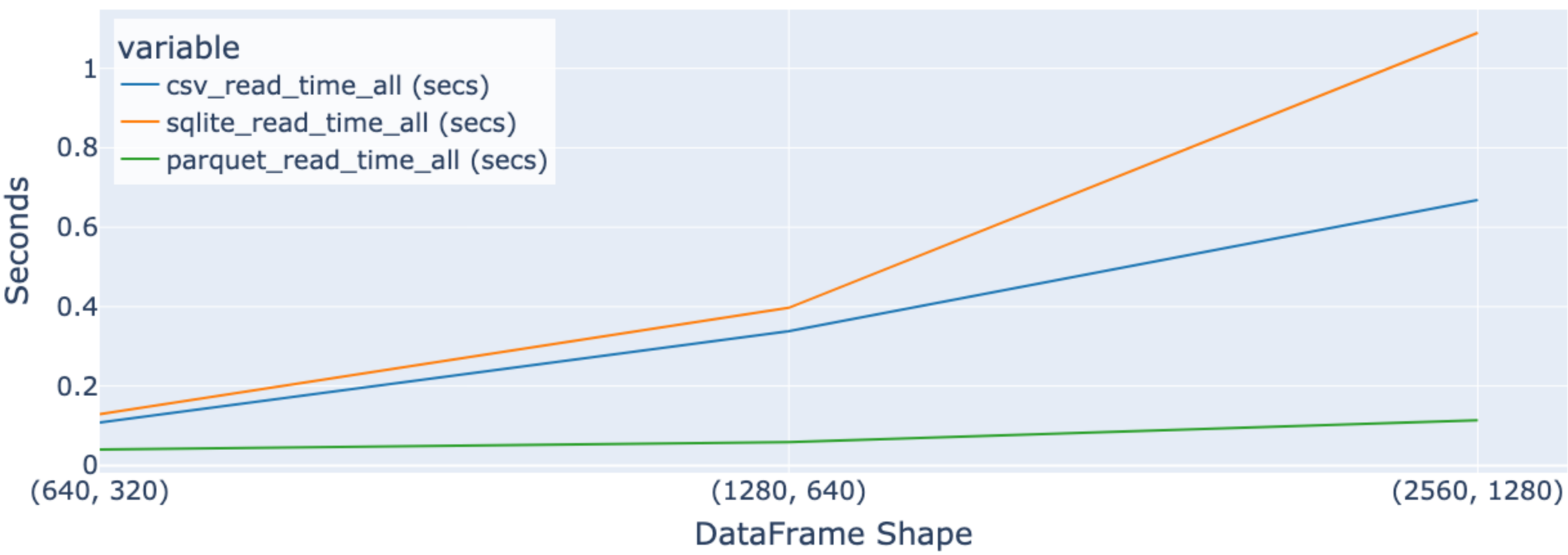


Figure 5. Chart showing read time durations for data from various file formats.

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.