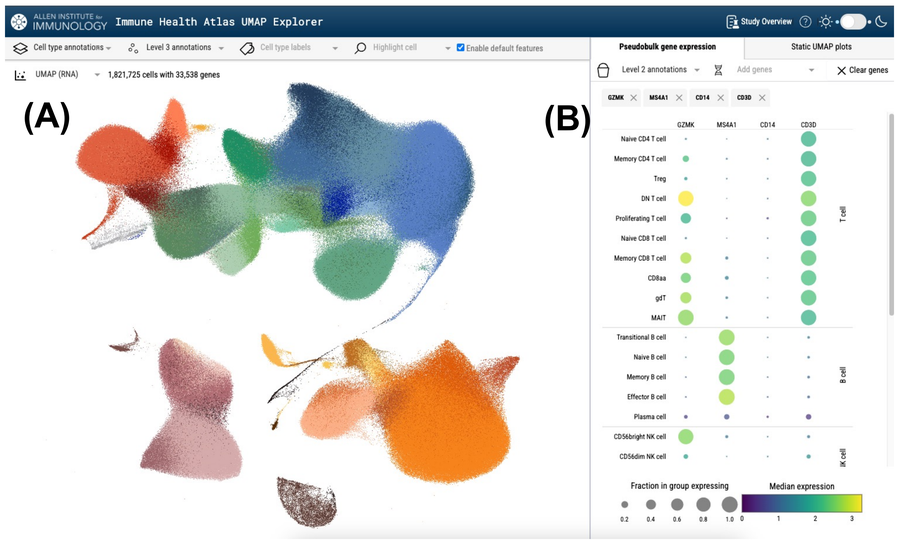
Advancing Open Science: Web-Based Applications for Real-Time Exploration of Large-Scale Omics Data

Christian M. La France, Lucas T. Graybuck, Vitalii Tereschenko, Yousef Aggoune, Madeline Ambrose, Aldan Beaubien, James Harvey, Nicole Howard, Neelima Inala, Ed Johnson, Autumn Kelsey, Melissa Kinsey, Jessica Liang, Paul Mariz, Stark Pister, Sathya Subramanian, Anne Vetto, Zachary J. Thomson, Peter J. Skene, and Paul Meijer

Overview

Scientific research reaches a broader audience when it is open, transparent, and accessible. To facilitate open data access the Allen Institute for Immunology has developed a framework which enables bioinformaticians to deploy Python-based web applications to visualize large-scale data. This includes a single-cell RNA sequencing UMAP explorer for real-time interactive visualization of millions of cells. Leveraging next-generation file formats, WebGL rendering, and optimized algorithms, this UMAP Explorer dramatically increases scalability while maintaining ease of development. This app is deployed publicly with our **Immune Health Atlas** (1.8M cells with 33K genes from 10x Genomics 3' scRNA-seq).



UMAP Explorer: Screenshot of the UMAP Explorer with the Immune Health Atlas which displays **A)** an interactive UMAP plot with >1.8 million cells and >33,000 genes. **B)** Pseudobulk gene expression plot displaying median expression of each gene and fraction of cells in each group expressing each gene.

Tools

Data storage

Zarr — Chunked and compressed multidimensional arrays optimized for for cloud environments with many concurrent users

Google Cloud storage of Zarr (up to hundreds of GB per app)

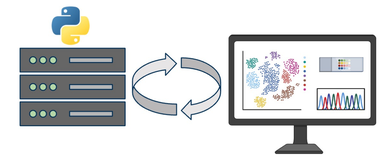
App components

plotly — Plotly Dash handles server-side code in Python for app structure and data management

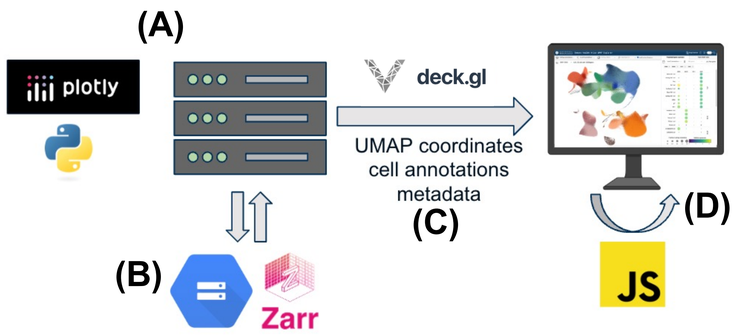
Plot rendering

deck.gl — WebGL rendering of up to 6 million points by offloading rendering tasks to the client GPU, reducing server trips and browser memory usage

Technical improvements

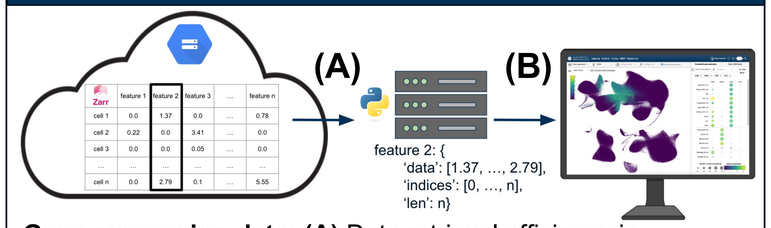


R/Python web frameworks typically favor ease of development over scalability, often causing frequent server trips, redundant data retransmissions, and requiring all data to be loaded into memory, resulting in poor scalability.



Schematic of UMAP Explorer: The Python server running a Plotly Dash app **(A)** retrieves data from Zarr in cloud storage **(B)** and sends it to the browser only once when needed **(C)**. Updates to the app that do not require new data are handled in the browser **(D)**. This simple restructure removes redundant trips to the server leading to a more responsive UI and increases scalability from ~200k points to ~6 million points.

Data Structures



Gene expression data: (A) Data retrieval efficiency is prioritized over storage efficiency using a Zarr dense matrix. **(B)** Once expression data are retrieved for a selected gene, converting the 1D expression array to a sparse array is fast and reduces data sent to the browser. For 1.8 million cells, plotting gene expression takes about 3-5s, vs 8-10s for a sparse matrix.

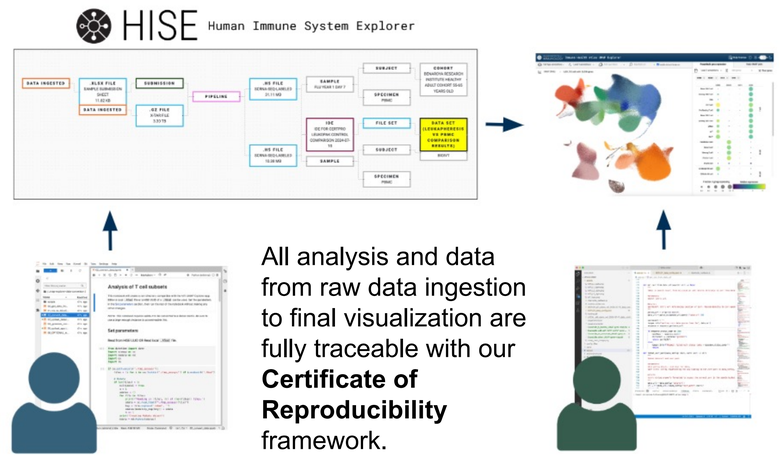
Cell annotations: Text annotations for each cell are encoded as integers then compressed with run-length encoding. This decreases the amount of data sent over the network and is fast to decode in the browser for plotting. These data are sent to the browser once per session when the app loads initially.

Original data (~65MB sent over network with each plot update):
["T cell", "T cell", "B cell", "Monocyte", "Monocyte", "Monocyte", … cell n]

Encoded data (~8MB sent over network once per session):
Code: {0: "T cell", 1: "B cell", 2: "Monocyte", … n: cell n}
Run-length encoded: [0, 1, 2, … cell n]
N occurrences: [2, 1, 3, … cell n]

Integration and traceability

This hybrid approach allows for the simplicity of Python development with the scalability of a custom-made web app. The app can be developed and maintained by a bioinformatician without extensive front-end development expertise. The app is easily deployed in our data platform **HISE (Human Immune System Explorer).**

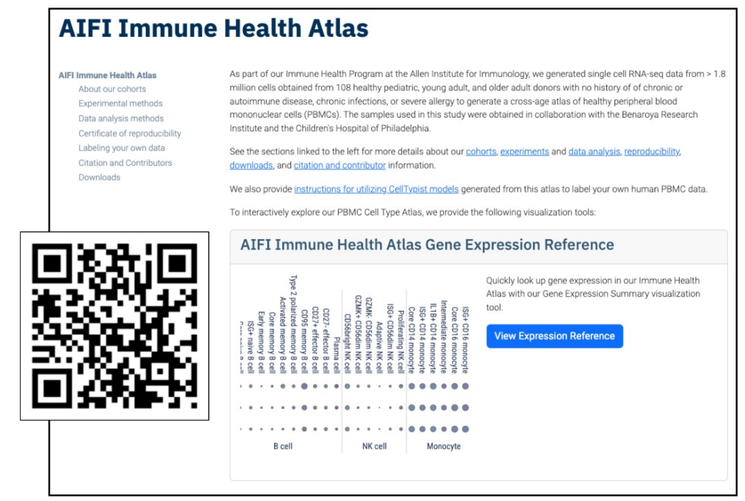


All analysis and data from raw data ingestion to final visualization are fully traceable with our **Certificate of Reproducibility** framework.

Certificate of Reproducibility
Meijer P, Howard N, Liang J, Kelsey A, Subramanian S, Johnson E, *et al.* Provide Proactive Reproducible Analysis Transparency with Every Publication. arXiv [cs.CE]. 2024.

Conclusions

These improvements significantly increase the scalability of large data apps in a form that can be maintained by a bioinformatician. Additional apps for new modalities are on the way! **Explore data and apps from the Immune Health Atlas here:**



*Apps work best on a desktop computer. Mobile performance is variable.

For more data and apps, visit: explore.allenimmunology.org

We wish to thank the Allen Institute for Immunology founder, Paul G. Allen, for his vision, encouragement, and support.