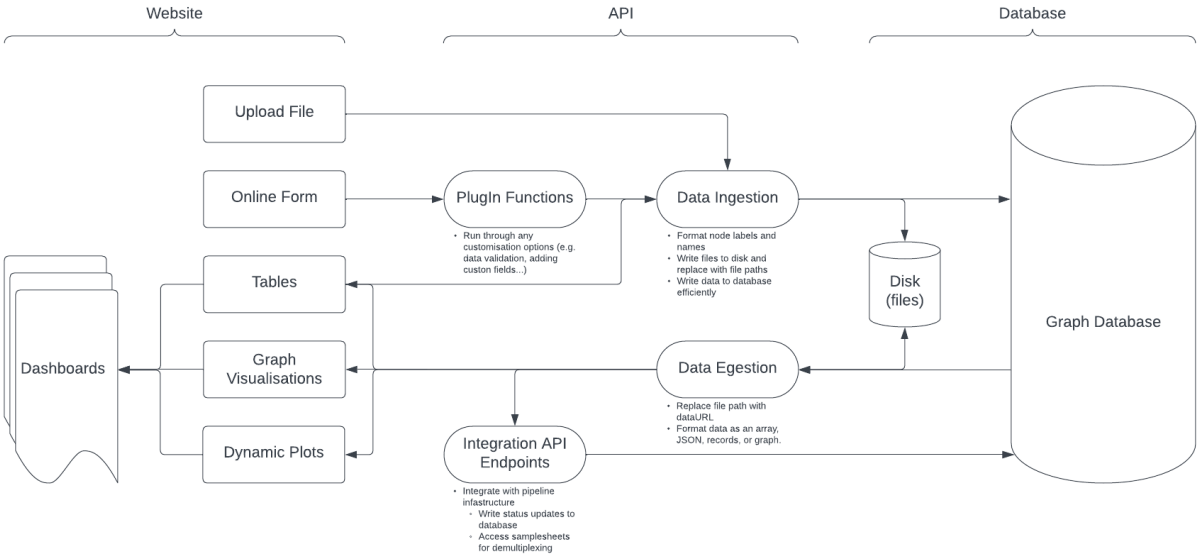
Metahub: A Simple and Scalable Open-Source Framework for Biomedical Metadata Management.

Abstract:

Effective management of metadata is crucial for advancing biomedical research and accelerating discoveries. However, existing solutions often fall short in providing a simple, scalable, and cost-effective way to organize and leverage metadata in a research environment. Existing solutions, such as Laboratory Information Management Systems (LIMS), are limited as they are cumbersome to set up, high maintenance, and expensive. To address these issues, we present Metahub, an open-source framework that provides a simple, lightweight, and easy-to-use solution for organizing metadata.

Metahub utilizes a website, an Application Programming Interface (API), and a graph database to ingest and analyze heterogeneous, constantly changing metadata. This framework scales across an organization and can incorporate a panoply of data modalities in an as-needed fashion. With Metahub, scientists, engineers, and analysts can efficiently store metadata, streamline data analysis workflows, and identify systematic errors in high-throughput experiments.

Moreover, Metahub's open-source nature and dynamic schema enable seamless integration with any project, compute environment, or software, while reducing the cost and complexity of metadata management. By providing a scalable, flexible, and cost-effective solution for managing dynamic and heterogeneous metadata in large-scale biomedical research projects, Metahub empowers researchers to focus on discoveries rather than data management.



Introduction

The Minimum Information for Biological and Biomedical Investigations (MIBBI) project established guidelines for capturing metadata in scientific research, highlighting the critical role that metadata plays in ensuring the reproducibility and transparency of scientific findings (1). Being able to provide and leverage metadata to support scientific insight is a fundamental pillar of high-impact research. The Tabula Sapiens project exemplified the challenges and significance of managing metadata in large-scale biomedical research projects (2). With half a million cells sequenced from fifteen subjects across five separate institutes, tracking patient phenotypes, sequencing methods, and reagents proved to be a challenge (2). Metadata was lost in communication, leading to roadblocks in data interpretation and analysis. Initial attempts to use Google Sheets and Google Forms to handle metadata failed to scale, were error-prone, and maintenance-heavy. Existing LIMS products were found to be too specific and structured to accommodate metadata from the diverse ongoing research at the Biohub. The most significant pain point was engineers and scientists lacking bandwidth to maintain and effectively utilize a LIMS to organize the project. To address these issues, we developed Metahub. The Metahub framework consists of a website, an API, and a graph database. Inspired by previous efforts to structure metadata for complex, cutting-edge research projects, Metahub leverages graph databases to provide a uniquely simple yet dynamic application capable of handling heterogeneous metadata.

Why Graphs?

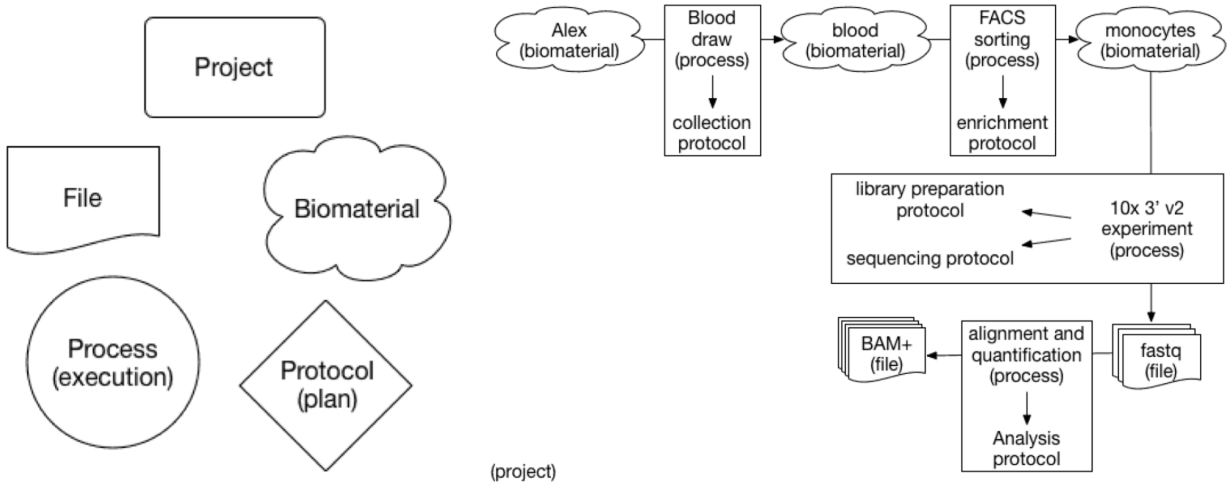
Graph databases are designed to handle complex, interconnected metadata. They store and query data as a network of interconnected nodes and edges, where nodes represent entities and edges represent relationships between them. Metadata is a fundamentally complex and interconnected type of data. Metadata fields can be represented as nodes, and relationships between fields as edges. The way metadata fields overlap across sample types, experimental types, and what matters for different projects is complicated and interconnected. Graphs are what allow Metahub to handle metadata simply yet competently, where other frameworks struggle without exhaustive maintenance. There are also significant computational advantages to developing with graph databases.

Graph databases are more performant for both querying and storing interconnected data. They use an optimized index-free adjacency model, which, together with graph traversal algorithms, leads to much faster and more efficient queries on complex data. Traversing edges and nodes in a graph scales linearly with the size of the graph, resulting in a complexity of O(n), while most SQL queries are logarithmic in nature with a complexity of O(log n) (3) (4) (5) (6). Modeling metadata in graphs is also more efficient than using tabular structures for storage. In tabular structures, data points are often stored multiple times in different tables, or values are repeated within a single table. In contrast, modeling data in a graph removes this redundancy by merging duplicate values into a single node. Graph databases can be further optimized with compression algorithms, such as GraphZip (7). Therefore, for handling interconnected metadata data, graph databases are a strong foundation for the Metahub framework.

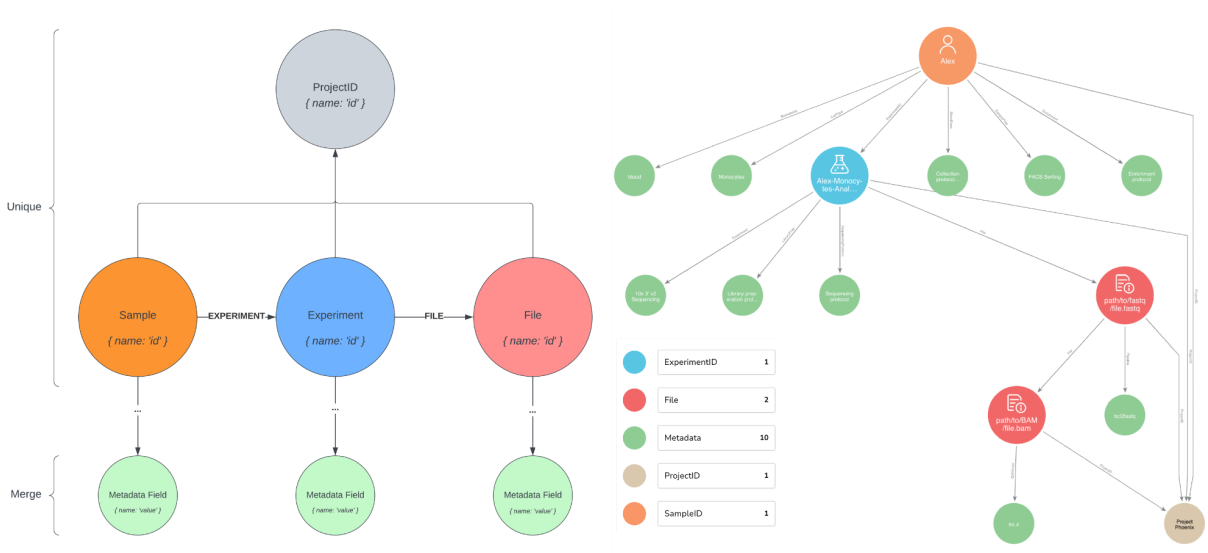
Neo4j is a popular graph database, with a free community version, that allows users to store and query highly connected data efficiently. It provides a powerful query language called Cypher that enables users to perform complex graph queries. Neo4j is widely used in industries such as healthcare, finance, and social media for applications like fraud detection, recommendation engines, and network analysis. It is a well validated, easy-to-use technology, which makes it strong for developing this proof of concept project.

Developing a Global Metadata Schema

To ensure the efficacy of the Metahub Framework, it is crucial to establish an optimal model for representing metadata. The Human Cell Atlas project demonstrates an excellent model of structured metadata (8). The human cell atlas organizes metadata around five major entities that are linked to one another, and all are self-describing, interlinking JSON entities. The principles guiding the schema organization are synonymous with the goals of the Metahub project (9). The limitation being, the Human Cell Atlas metadata schema requires a large team of dedicated engineers and data wranglers to maintain this database. Furthermore, there is no way for non-technical users to analyze this metadata.



Metahub adopts some of the same principles from this metadata schema, adapted for a graph database structure. There are three core entities around which metadata is organized: the sample, the experiment, and the file. Other metadata fields are then connected to these three 'index' nodes, with the field set as the label and the value set to the name. This allows for a highly flexible schema that is still structured for application logic and analysis. Certain metadata fields, such as ProjectID, can be used to group samples, experiments, or files from a specific project. Duplicate field/value combinations are merged into a single node, while all 'ID' values are unique.



This schema is flexible to any incoming requirements, while still remaining structured enough for applications to be developed around it and graph theory analysis to be performed. Duplicate nodes are merged, tying together samples, experiments, and files with overlapping

metadata. This simple artifact of modeling the data in a graph already clarifies how experiments and datasets may overlap and inform each other. Structuring metadata in this way, where it is possible to directly observe how quality control metrics are connected to the way samples are collected and prepared, or the libraries and methods used in an experiment is hugely valuable for scientists. The potential returns for investigating interconnected metadata through graph analyses are still unexplored.

Interfacing with Metadata

For graph databases to be effectively leveraged by researchers, user interfaces must be developed. Developing applications on this type of graph database poses a challenge, as any API's or website's design must be highly dynamic in order to take advantage of the graph schema and ensure maintenance of the framework is minimal. For developing the Metahub project, there are three main user stories:

- **Leadership:** An overview of research, ability to search and explore metadata.
- **The Scientist:** Facilitate collaborative projects and work, ability to control metadata ingestion for specific metadata fields or individuals, tracking metadata and statuses for specific projects or technology platforms.
- **The Engineer:** Programmatically read and update metadata for third-party applications.

To handle metadata ingestion, we designed two workflows. The first workflow supports collaboration and structured metadata ingestion. Embedded in our website, we developed a no-code website editor using TinaCMS that allows scientists to design online forms for users to submit samples alongside relevant metadata (10). These forms propagate the sample metadata to the backend, where engineers can easily implement custom validation and processing in a module before the metadata is uploaded as a 'Sample' to the graph. The scientist can then track and manage the status of submissions through a dashboard in an internal page on the website. This dashboard also renders dynamically computed plots and visualizations of the metadata. Samples are grouped by the name of the online form used to submit them.

The second workflow is similar; however, metadata is ingested through a user uploading an excel file on a separate page containing metadata specifically formatted so it can be converted into a graph. This allows users to upload bulk metadata quickly, and researchers who are comfortable working with spreadsheets can continue using their preferred tools. In both workflows, the metadata is verified and processed by a backend API, which is responsible for validating and structuring the metadata into the graph schema.

To facilitate easy access to the metadata, we developed a search page that displays results based on the user's search query. The search page is capable of handling queries with multiple layers of specificity, such as searching for samples from a specific project and with certain experimental parameters. This search functionality empowers researchers to find the data they need and explore the underlying metadata relationships.

To address the needs of engineers, we developed a GraphQL API that enables programmatic access to the graph database. This API allows engineers to build applications on top of the Metahub infrastructure, making it easy to integrate other software or tools into the ecosystem.

Facilitating projects at the Chan-Zuckerberg Biohub

The first workflow for handling sample submissions is utilized heavily by the genomics platform. Genomics sequence a large number of samples per day from internal users and collaborators, so require a robust and effective system for validating which submissions are allowed and ingesting then tracking the ones that are. Scientists were able to effectively use the no-code editor, and submissions dashboard to organize submissions. Pipeline engineers were able to access the integration API in order to automate demultiplexing pipelines.

The second workflow successfully ingested metadata from excel files, which supported both a bioengineering and developmental biology research project respectively. This system allowed users to collate and store metadata for later use (not fully implemented).

Future Directions

The Metahub project demonstrates a successful marriage of graph databases and metadata management in the realm of scientific research. The schema's flexibility and the user interfaces make it a practical tool for researchers to collaborate and access data. In addition, the underlying graph structure offers novel opportunities for metadata-driven analysis and discovery.

The possibilities for future development are vast. One potential avenue is integrating machine learning algorithms to predict relationships between metadata, experiments, and samples, helping researchers uncover new insights. Additionally, developing more advanced visualization tools that leverage the graph structure can further enhance the user experience and facilitate data exploration.

In conclusion, the Metahub project demonstrates the power of graph databases for managing and analyzing metadata in scientific research. By developing user-friendly interfaces and a flexible, yet structured schema, we have created a system that promotes collaboration, data discovery, and novel analyses. The potential impact of this approach on the scientific community is immense, opening the door for more efficient research and breakthroughs in various fields.

Acknowledgements:

References:

1. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2771753/
2. https://www.science.org/doi/10.1126/science.abl4896
3. https://arxiv.org/abs/1004.1001
4. https://www.arangodb.com/2016/04/index-free-adjacency-hybrid-indexes-graph-databases/
5. https://www.arangodb.com/2018/02/nosql-performance-benchmark-2018-mongodb-postgresql-orientdb-neo4j-arangodb/
6. https://neo4j.com/news/how-much-faster-is-a-graph-database-really/
7. https://arxiv.org/abs/1703.08614
8. https://data.humancellatlas.org/metadata/structure
9. https://data.humancellatlas.org/metadata/rationale#design-choices
10. https://tina.io/