LabBook: Metadata-driven Social Collaborative Data Analysis

Eser Kandogan, Mary Roth, Peter Schwarz,
Joshua Hui, Ignacio Terrizzano
IBM Research -- Almaden
San Jose, USA
{eser, torkroth, pschwarz, jhui, igterriz}@us.ibm.com

Christina Christodoulakis*, Renée J. Miller*
Department of Computer Science
University of Toronto
Toronto, Canada
{christina, miller}@cs.toronto.edu

Abstract— **Open data analysis platforms are being adopted to support collaboration in science and business. Studies suggest that analytic work in an enterprise occurs in a complex ecosystem of people, data, and software working in a coordinated manner. These studies also point to friction between the elements of this ecosystem that reduces user productivity and quality of work. LabBook is an open, social, and collaborative data analysis platform designed explicitly to reduce this friction and accelerate discovery. Its goal is to help users leverage each other's knowledge and experience to find the data, tools and collaborators they need to integrate, visualize, and analyze data. The key insight is to collect and use more metadata about all elements of the analytic ecosystem by means of an architecture and user experience that reduce the cost of contributing such metadata. We demonstrate how metadata can be exploited to improve the collaborative user experience and facilitate collaborative data integration and recommendations. We describe a specific use case and discuss several design issues concerning the capture, representation, querying and use of metadata.**

Keywords- metadata; collaboration; data discovery; data analytics

* Supported by NSERC and NSERC BIN.

I. INTRODUCTION

Much of the current research on solving large-scale data-intensive problems has focused on the development of algorithms and systems that facilitate the processing of very large data sets. However, another important characteristic of these data-intensive problems, and one that has received far less attention, is that solving them typically requires the pooling of data, tools and expertise from multiple disciplines. In science and in business, therefore, demand is increasing for open platforms that support the sharing and collaborative exploration and analysis of data. In the scientific community, open science initiatives aim to accelerate scientific discovery by making experimental data and lab notes accessible and re-usable by the broader community [1]. For example, in neuroscience [2] and in bioinformatics [3], open platforms are being built to understand neurodegenerative diseases and the bacterial diversity of cities, respectively. Likewise, within an enterprise, data is often re-used, re-combined and re-cycled by many different groups to answer a wide range of business questions. Studies of business intelligence specialists [4][5] suggest that analytic work in an enterprise occurs in a complex ecosystem of people, data, and analytic tools collectively working together. These studies point to sources of friction within this ecosystem, including data integration, expertise finding, reuse, and interoperability.

In science and in business, supporting collaboration among a diverse set of users is not easy. Differences in information needs and goals make the utility and trustworthiness of data, tools, and people vary significantly from one community to another. Differences in language and terminology can also prevent effective collaboration and re-use. Studies report that business analysts are needed to bridge the language gap between business and technical people, transforming business objectives to database queries [4]. There is also significant diversity in the structure, type, volume, velocity, and veracity of data. Inconsistencies in schema and formats may cause time to be spent on data wrangling before data can be integrated [6]. Likewise, there are many analytic tools and systems, requiring time to be spent orchestrating the flow of data from one tool to another to create a comprehensive and cohesive analysis. Furthermore, data, analysis techniques and people are constantly evolving. Schemas evolve, even for datasets released regularly such as census and labor statistics. New data supersedes old, hypotheses need to be revised, and past conclusions may be invalidated. New people join in, bringing different perspectives to projects, while the interests of experienced members change over time. Lastly, data privacy and sensitivity considerations necessitate data governance, and the scope of collaboration must be carefully controlled. Broadly, these challenges can be summarized as issues related to *diversity of resources* (i.e., people, data, analytic tools) in a *dynamic evolving analytic ecosystem,* requiring *iterative, collaborative* approaches with *contextual, personal,* and *governed* delivery of resources.

In this paper, we argue that the key to addressing these problems is metadata. If we are able to collect more metadata about people, data, and tools, then we can put together the right data, people, and tools to solve complex analytic problems. Such metadata should include schematic metadata (i.e., how data is structured in files, tables, columns), as well as semantic metadata (e.g., the meaning of data attributes and values, relationships between datasets), collaborative metadata (e.g., who collaborates with whom, on which data, using which tools), and contextual metadata (e.g., thoughts,

978-1-4799-9926-2/15/$31.00 ©2015 IEEE

hypotheses, decisions as people work). The metadata may be previously modeled metadata (for example, the schema of a structured or semi-structured dataset) or it may be inferred using a knowledge discovery process (for example, using a semantic annotation process to tag a dataset with concepts from an ontology or by extracting semantics from unstructured documents). The metadata must be complemented by a user experience design that facilitates seamless and silent capture of metadata as people interact with data, applications and each other, while also simplifying the dissemination of analytic output. The system should support people working with a multitude of tools, curating and enriching data over time, developing models, creating visualizations, leveraging each other's work and making and revising data-driven decisions in an incremental fashion. The system must have an open architecture into which analytic tools and applications integrate easily, and provide a central repository from which the tools can consume data and metadata and to which they can contribute back new data and metadata in turn. Such a repository may constitute a knowledge resource in and of itself, leading to an improved understanding of how people work with data and make decisions. This knowledge resource can be exploited to provide a personalized user experience, make contextual recommendations of people and data, and identify and reuse best practices.

The contributions of this paper are: (a) design and implementation of a *metadata schema and repository* for open analytics platforms as a queryable property graph, which represents schematic, semantic, collaborative, and contextual metadata; (b) design and implementation of a *social collaborative user experience* that exploits such metadata and reduces the cost of contributing metadata; (c) design and implementation of an *open analytics architecture* that serves metadata to analytic applications; and (d) demonstration of several *uses of metadata* in data integration and data recommendation.

The rest of the paper is organized as follows. First, we present a real use case that informed the design of our work. We then describe LabBook, our implementation of an open social collaborative analytics platform, and describe its metadata repository, social collaborative user experience, and open architecture. Next, we describe how LabBook uses metadata to support analytic work including data integration and recommendations of data and people. We then briefly cover related work in the fields of metadata management, open science and data marketplaces. We conclude by discussing several issues relating to the capture, representation, and querying of metadata.

II. USE CASE: METAGENOMICS

The recently-announced Consortium for Sequencing the Food Supply Chain (SFSC) will use metagenomics to study the genetic diversity of microbiomes in the food supply chain and create a corpus of data that reflects a metagenomic understanding of ingredients and their microbiomes [7]. LabBook is designed to serve as the collaborative analytics platform for members of the consortium. We interviewed several members of the consortium (three bioinformaticians and a computer scientist) to inform the design of LabBook.

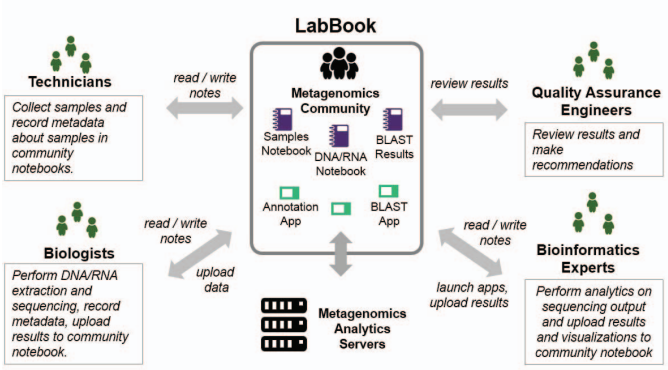

Figure 1. Metagenomics Use Case

The consortium is made up of members with diverse backgrounds, skills, and expertise, including biologists, machine learning experts, computer scientists, factory workers, and executives (Figure 1). Technicians take samples and send them to biologists. The samples are prepared and run through DNA sequencers, producing output files that contain sequence databases. These files are then shared with computer scientists, who implement algorithms developed by bioinformaticians. The algorithms compare the sequences with known reference data to characterize the microbiome of the sample. All participants iteratively and collectively analyze the results with the aim of detecting anomalies that might indicate the presence of a contaminant.

We note three important characteristics of this scenario. First, other than a shared goal, the constituents do not share the same level of expertise or even a common set of tools. Second, the scale of the data and of the analytics is significant, thus rendering email quite ineffective as a means for collaboration. A single dataset may span 10 terabytes of data contained in over 40000 files, and each file may require different processing and analysis. Finally, because this is a scientific discovery process, both the process and the outcome of the collective analysis are undetermined *a priori* and, in fact, both represent goals of the consortium; only in hindsight will the collaborators understand that they have discovered something important, and therefore the provenance of that discovery is key to establishing the reference data and a reproducible process.

Bioinformatics is a field rich with a variety of sub-specializations. Such an environment breeds a *'loose'* collaboration style, often starting face-to-face at a conference, with conversations like "*this is what I can do for you, then we'll reconvene in a month or so.*" Another consortium member spoke of a pattern of recent projects, and described them as "*hub and spoke*", where "*collaboration is one to one, everybody is independently collaborating with the PI (Principal Investigator), who is sort of doing a crowdsourced analysis of his paper by sharing his data. He is a social networker, everybody has special takes on data and if lucky, it becomes a paper or a paragraph in the paper.*" In such a collaboration style, the interaction typically follows a pattern like "*I produce some analysis, generate a plot, attach it to an email, say, 'Take a look at this', He says,*

'Oh, that is interesting', I ask, 'Do you have this data?', He tells me to go to this ftp site...' This is what a lot of science is like today..." Such loose collaboration is not supported very well by current collaboration tools, as one must spend a considerable amount of time to understand the context, datasets, and work done to respond appropriately. As one of the bioinformaticians put it: "*Email is not good. I need to invest a lot of effort in reading the email and trying to understand sort of what the context of this plot is, what do I know about this data, … Often I say, I guess I will look at it later. There are something like 40,000 datasets!*"

Such complexity in the diversity and scale of the data is at the heart of bioinformatics: "*Everything is a dataset. The thing you upload is a dataset, then if you process it and output something, that is a dataset. If you are looking at any particular dataset, you may want to know where it comes from*." The data represents a jigsaw puzzle, which is generated by a trial-and-error process that identifies and stitches together larger and larger genome fragments in order to create a complete bacterial genome. Besides data that is privately created and reused by bioinformaticians, there is also a large body of public data available. For example, the National Center for Biotechnology Information (NCBI), is an "*ecosystem that defies understanding at first glance. It is the work of hundreds of people over two-decades or more*." At such scale, people "*only understand a slice, kind of play with it, and experiment.*" For areas like bacteria, this is a very dynamic ecosystem with new data arriving weekly. Privacy is important, even when data is de-identified, as it may still contain traits of illnesses, revealing clinical details.

Bioinformaticians use a wide variety of tools for very specific tasks. They use wet labs to prepare samples and record metadata. Machine learning experts use various programming tools and scripting languages to develop workflows that implement specific sets of analyses over the output of sequencing. As one of the bioinformaticians put it, "*The community of bioinformatics is entirely built on software produced in academic labs, written by grad students. There are thousands of tools out there, and new tools come out all the time for the latest datasets.*" In such a diverse tool space, people have their own favorite 10 specific tools, while a few tools have the "*exalted status that 99% of people use.*" Each part of the work entails use of special tools, and work is done by "*putting different blocks together to come up with the process you want.*" Work can sometimes be repetitive, and may require digging up old emails to find out about how work was previously performed. Recalling a recent project, one of the bioinformaticians said that: "*They continuously get new data and update the references. Whenever they have a new reference we have to repeat all this mapping again*." Communication issues may also cause problems: "*We didn't understand from the start all their experimental setup. It was never communicated enough what data we should actually be comparing.*"

In summary, the metagenomics use-case exhibits many of the characteristics that argue for an open data analytics platform: it presents a dynamic living ecosystem with a continuous stream of datasets, people, and tools entering and exiting the collaboration. People have a variety of backgrounds, use a variety of tools, and have a variety of goals to achieve. Thus, there is no one authoritative source for data, no *de facto* standard for tools, and no well-established processes for analysis. This not only necessitates an open data analytics platform, but also incremental solutions to challenges such as personalization, collaboration, provenance, and reuse.

III. LabBook: Graph, User Experience, Architecture

LabBook is an open, social, collaborative data analysis platform designed explicitly to accelerate discovery by reducing friction in the analytic ecosystem, i.e., by helping users to quickly find relevant data, collaborate with others, and reuse analytic work within a community. In addition, LabBook provides a seamless and transparent provenance mechanism. A core element of LabBook is a *metadata graph,* which captures the interactions between data, people, and analytics and is used to facilitate, for example, recommendations of data and people related to a project. Additional metadata is silently contributed back to the metadata graph as people use the *social collaborative user interface,* in which events trigger the addition of new nodes and relationships to the graph. LabBook's *open extensible architecture* allows integration with many different analytic applications, thus serving as a general knowledge resource that can be exploited to provide a contextual and personalized user experience, to help users keep up-to-date with current work in their community, and to recommend datasets, people, and visualizations. Below, we describe LabBook's (1) metadata graph, (2) social collaborative user interface, and (3) open extensible architecture in more detail.

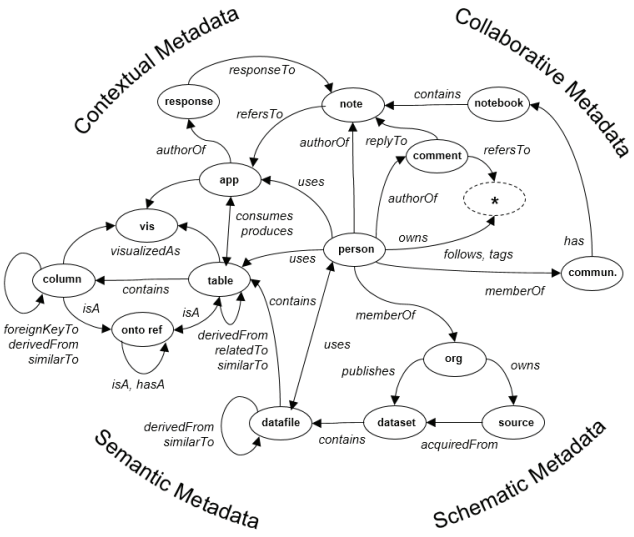

Figure 2. LabBook graph brings together schematic, semantic, collaborative, and contextual metadata as a single resource

A. Metadata Graph

Broadly, the property graph contains entities and relationships that capture schematic, semantic, collaborative,

and contextual metadata. *Schematic* metadata describes how data is structured, e.g., a zip file that may contain multiple CSV files, each containing multiple tables and sets of columns, is represented in the graph via entities such as dataset, data file, table, and column, and via containment and association relationships. *Semantic* metadata captures the meaning of data, e.g., that a table represents information abuot bacteria, or that a column represents a bacterial habitat, and also how data items are related, for example, the derivation relationship between tables. *Collaborative* metadata describes relationships between people, e.g., people following each other or collaborating in communities. Finally, *contextual* metadata captures the context of data use, e.g., scientists sharing their thoughts and hypotheses in notebooks where they can respond or contribute ideas.

The current graph schema (Figure 2) includes various entity types. All entities have basic attributes such as *preferredLabel*, *description*, *tags,* etc. and relationships such as *ownedBy* and *hasAttachment*. Several subclasses of **Agent** represent users, the organizations they belong to and the relationships among them. The **Asset** class represents resources, with subclasses such as **Dataset**, **Datafile**, **Table**, and **Column**. The **Application** class represents executable tools and services and provides relationships such as *consumes,* and *produces*. A **Visualization** is a visual representation of a data asset and a **Notebook** represents a collection of notes authored by one or more people. The **Notes** can be simple free-form text, or input for launching an app with parameters, and may themselves refer to other entities. A **Response** represents structured output from an application invocation represented by a **Note**. Lastly, a **SemanticType** represents a reference to a semantic hierarchy, such as a bacterial taxonomy.

Like entities, relationships can also have attributes. For example, the *similarTo* relationship between two assets (e.g. tables, columns, datafiles) has attributes such as *strength*, identifying the quality, and *source,* identifying the origin of the relationship (e.g., the name of a data service). The graph also supports custom user-defined attributes and relationships, which are important for external services that contribute to the graph.

The metadata graph is implemented using the Titan graph database [8], with Apache Cassandra [9] as the backing store, and exposes a set of services via RESTful APIs. These services are used to query, search, and monitor the metadata graph and also provide statistics over the graph. The query API supports graph traversal and update operations, facilitated by Gremlin [10], a functional graph language that allows complex queries to be constructed by chaining operators as path-expressions. The search API, implemented using Elasticsearch [11], supports keyword-based search with facets for type, date, and user-defined attributes. The search API also allows one to easily pull in entities related to those that match a search term. For example, a **Person** entity can be indexed not only in terms of its attributes, but also in terms of attributes of related entities such as communities, notebooks and datasets the person owns. Thus, a search for a dataset name also returns all users, notebooks, and other entities related to the matching datasets. The monitor API allows real-time tracking and searching of low-level graph events, including entity updates and relationship additions and removals. The activity API is built on top of the monitor API to support aggregation of all events related to a person, facilitating awareness and notifications. Finally, the statistics API provides statistics over the graph that are used, e.g., in recommenders that calculate the score of a dataset based on the number of notebooks that refer to it.

In open systems, governance and access control are key requirements. Thus, the LabBook graph APIs include both access control and authorization services. The access control service governs access to individual entities in the graph, for example, restricting the visibility of a dataset, or restricting the invocation of an app to selected people. The authorization service tracks and enforces data licensing agreements. For example, when access to a particular dataset is requested (either directly as download, or indirectly as a visualization), the service verifies that the user has agreed to the relevant terms and conditions.

The metadata graph is evolving and growing continuously, and can be populated in numerous ways. Information about assets such as datasets or applications is either provided by the user or extracted automatically. For example, when a dataset is imported from data.gov, basic metadata from the source is also imported. Applications can also populate the graph. For example, a metagenomics app publishes the dataset resulting from analysis as a new derived dataset. Most importantly, use of LabBook silently evolves the graph -- as individuals interact with entities, new entities and relationships are added to record that activity.

B. *Collaborative User Experience*

The LabBook user interface design delivers a social, conversational user experience, where each user has a homepage, can create communities, can follow entities such as people and datasets, and can explore data in an *ad hoc* and agile manner (Figure 3). A user's homepage lists her communities, recent notebooks, frequently used apps, datasets, and documents, as well as recommendations of people, data, notebooks, and applications based on her

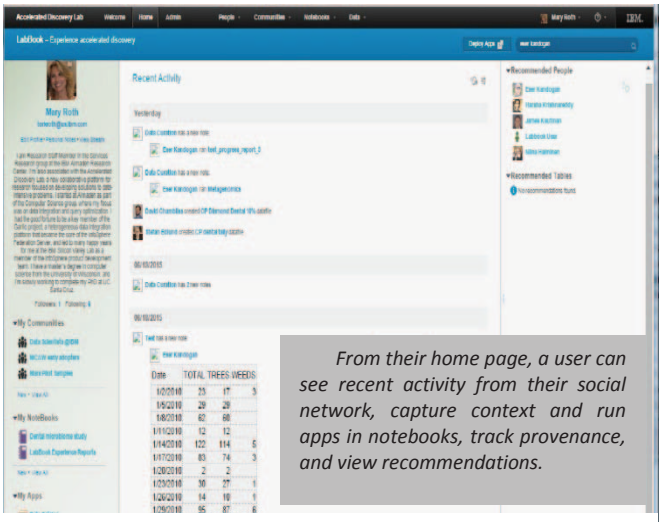

Figure 3 LabBook Social Collaborative User Experience

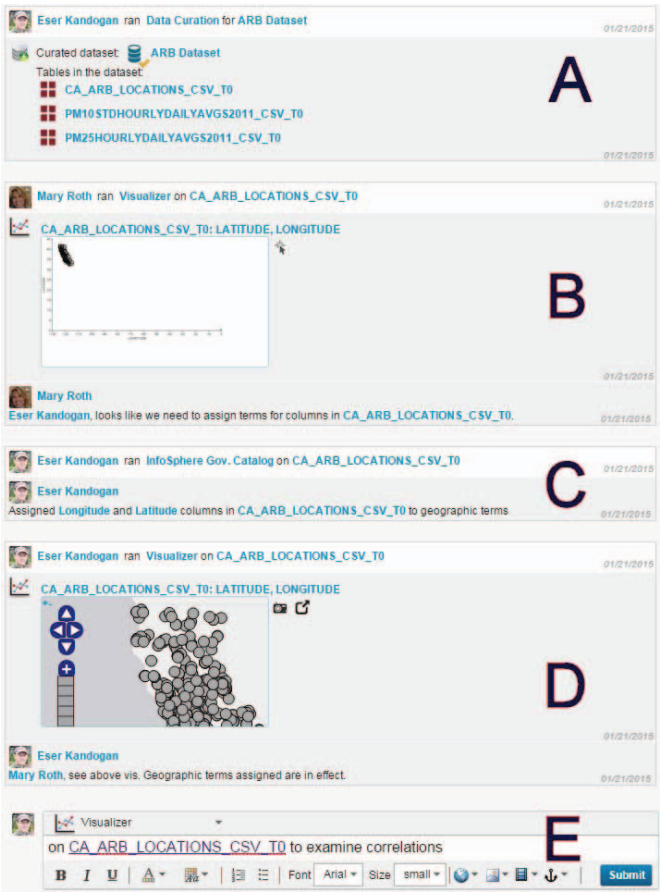

Figure 4. In a series of notes, user (a) curates and adds tables, (b) creates an initial visualization, (c) enriches data by assigning columns to ontological references, (d) produces a map visualization that leverages such semantic metadata, and (e) and is about to launch the visualizer app to examine correlations.

activity. At the center of the homepage, other activities relevant to the user are summarized, including notes mentioning the user, updates to notebooks of which she is an author, new notebooks added to her communities, and new releases of datasets she is following. The activity stream gives a quick overview of recent activity, but also provides an opportunity to make quick responses to urgent notes.

While the activity stream allows the user to respond to multiple ongoing activities quickly, the user can also switch to a specific community page or a notebook to do focused work. Any entity in the UI is clickable, allowing the user to switch to a page designed specifically to show an entity and its related information. For example, a dataset page shows lists of data files and tables, visualizations, users, and comments, along with schematic metadata such as data sources and publishing organizations. A community page contains people and notebooks, along with apps and datasets relevant to that community. For example, the page for a community of genomics researchers might include biologists and machine learning experts, DNA sequence datasets, specific genomics apps, and notebooks for discussing the metagenomic content of a sample.

Notebooks act as a digital version of a scientist's lab notebook. They contain free-form notes about work being conducted, such as thoughts regarding analysis of a particular dataset, and artifacts that would facilitate such analysis, such as visualizations and models. Notebooks can be public, private or shared, allowing information to be compartmentalized if necessary. By capturing the exchange of ideas, knowledge and expertise in notebooks, LabBook facilitates collaboration among a community of individuals Apps can also be invoked from notes in the context of a notebook, capturing the input, output, and status of execution. For example, in Figure 4, Eser starts by uploading a zip file (ARB Dataset) that contains air quality measures for California, using a curation app to analyze the zip file and create three tables as a result. In the next step, Mary invokes a visualization app to explore the locations of sensors collecting the air data, passing a particular table as a reference. The visualizer app responds back by presenting a dialogue to let Mary select the specific columns to include in the visualization. It then renders the visualization back into the notebook, making it available to all notebook users. Mary adds a comment, indicating she was expecting the results to be displayed on a map. Next, Eser uses another app to associate the *Latitude* and *Longitude* columns with semantic tags. This information is captured in the graph, so when he reruns the visualizer, it can leverage the new information to choose a map as a more suitable rendering of the data.

Notebooks enable a key feature of LabBook by allowing collaboration to co-occur within the same context as the actual analytic work. This is in contrast to standalone collaboration applications, where users need to actively integrate work artifacts to support their collaborative use. Notebooks give users a complete and comprehensive log of their collaborative work, for both their benefit and for the benefit of their colleagues, for example, to examine and discuss how a particular analysis was conducted. Notebooks are a natural and seamless way to capture the context of how a particular result was derived, enabling governance and provenance of data assets. Connections in the metadata graph facilitate collaboration indirectly, by enabling reutilization of artifacts created by others (e.g., previously curated or analyzed datasets, visualizations) and directly, by connecting users with identified experts. For example, when a dataset is referenced in a notebook, another user can traverse from the dataset page to any notebook in which this dataset is used and also traverse to related individuals and communities. Such browsing capability allows the user to see the context in which the dataset was used, where it can be assessed for its trustworthiness.

Collaboration is also implicitly supported through people recommendations (see Section IV.B). These can be general, based on all of a user's interactions, but may also have a specific search as their context, in which case the recommenders are used to rank and personalize the search results. For example, when users are searching in a particular notebook, matching datasets owned by the notebook's authors will be ranked higher than others.

The graph API allows overlaying of connections from external social networks (e.g., Facebook™) on to the

LabBook graph. As an example, we currently pull such data from our corporate social network, IBM Connections [12]. Relationships among individuals are discovered by the Social Networks and Discovery (SAND) service [13], which looks for evidence of strong social connections based on co-authorship and other factors, and populates these relationships in the graph along with a score and date.

C. Architecture: Analytics Integration Hub

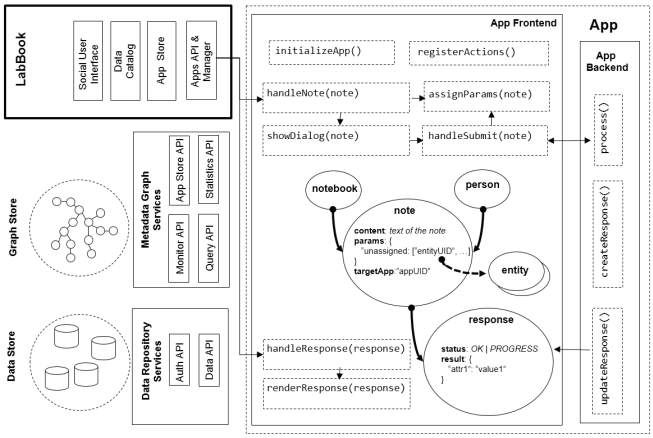

Figure 5. Apps Architecture and API

An extensible apps architecture is critical to achieving an open analytics platform, and especially so for the metagenomics use case, as bioinformaticians uuse a large set of analytic tools. LabBook's web-based apps architecture allows users to dynamically upload and register apps, and its app framework provides APIs that allow anyone to develop apps and integrate them. LabBook comes with several default apps for visualizing data, browsing catalogs, recommending data and people, etc.

A contributed app is packaged as set of source (e.g., JavaScript) and support files (e.g., images, UI templates, or internal data), along with a descriptor that lists package contents and app metadata (e.g., name, description, icons, applicable roles, etc.). Typically, contributed apps have frontend and backend components. The backend components, often deployed on another server as a web-service, do the heavy lifting in terms of computation and input/output, while the frontend components support basic user interaction and integration with the LabBook UI. Apps can leverage the services provided by the metadata graph's RESTful APIs, and can expose their functionality from different parts of the collaborative platform (e.g., in notebooks, and in context-menus) (Figure 5). The essential app-related APIs are: (1) the app store API, providing registry and catalog services for apps; (2) the query API, supporting queries over the metadata graph; (3) the monitor API, tracking graph events; and (4) the statistics API, for periodically computing general statistics over the graph. Additional APIS include a data API for retrieving content and an authorization API that allows apps to verify user access to data content. Using these services, apps can perform computation over the graph and put computed output (e.g., a model, visualization, or derived data) as a response into the graph and render it into a notebook.

Integration with the LabBook user interface is typically handled in the app frontend by extending the JavaScript Apps API, which facilitates the flow of interaction between the app and the user interface. When LabBook is loaded in the browser, JavaScript app code specified in the app descriptor is dynamically loaded and initialized. When an app is launched in the context of a notebook, the app receives the content of the note and parameters. It may then display UI dialogs to gather more information, or call backend APIs. The app may use the various data and graph services to fetch more data and continue performing the computation. Once the computation is done, the app creates a response, which is then linked to the originating note in the notebook. Upon receiving the response event, the LabBook UI calls the app to see if further interaction is needed to render output (based on the response) into the note. If the backend computation is a long-running process, the backend can occasionally update the response entity with the status of the computation, which may also be rendered into the notebook.

A key aspect of this architecture is that the graph serves as an integration hub. It is available to be used not only by the LabBook UI, but also by all the analytic apps. Thus, one can capture the context of all interactions between people and data as revealed through their invocation of various applications. As such, the graph serves as a general repository of how data is used, and how data is transformed from one analytics app to another, while capturing provenance information all along the way.

IV. LabBook: Exploiting Metadata

In this section, we illustrate how a unified metadata graph, which represents schematic, semantic, collaborative, and contextual metadata about people, data, and tools, can improve data integration and recommendations, and we argue that the combination provides a richer fabric for discovery than the sum of the individual metadata types.

A. Data Integration

An open analytics platform contains data on a wide range of topics, from a variety of publishers that is owned and curated by numerous users, thus providing a springboard for data integration [1]. Metadata management for data integration has historically focused on schema-based techniques to discover mappings or referential integrity constraints between datasets [14], content-based techniques to identify columns that contain overlapping data values [16], and semantics-based techniques [17] to discover data with similar meanings.

However, a challenge with such discovery techniques is that they run independently, making it difficult to derive real value. For example, knowing that two datasets contain columns whose values are formatted the same way and are in similar ranges offers a clue that the datasets may be combined. But, do the columns represent the same semantic concept, such as temperature? Do they contain the temperature of the same sample at the same time? Who created these datasets and for what purpose? All these factors

would need to be considered to decide whether it makes sense to actually integrate these tables. The value proposition of LabBook's metadata graph is that it serves as an integration hub where disparate metadata can come together and facilitate an iterative and collaborative approach to integration.

The LabBook metadata graph has several entities, attributes, and relationships intended to capture metadata for integration. While relationships such as *foreignKeyTo* represent data schema constraints, another relationship, *relatedTo,* can be used to indicate a content-based relationship. These attributes and relationships can be entered by the user, populated from a standard metadata repository or derived by an app. LabBook's curation app derives schematic and semantic metadata by analyzing the structure and content of files. If data is imported from an open-data site, additional available metadata, including a description, the source, the publisher, the number of downloads, etc., is captured by the app as attributes and relationships. Thereafter, social and collaborative metadata is collected over time through the LabBook UI and APIs, including end-user tags on columns and tables, taxonomic terms assigned to columns, tables, and datasets, or even schema mappings between tables. While at the onset a dataset might have a limited amount of metadata, additional entities and relationships may be created by schema mapping tools, linked data discovery services, and automatic techniques that use ontology resources. The graph provides a holistic ecosystem for data integration, capturing a temporal history that records all of a user's activities, including for example, explicitly combining datasets to create a visualization.

B. Recommendations

Presenting tailored content to the user can make for a more productive user experience. To facilitate collaboration among researchers, LabBook provides its users with recommendations for various types of assets, including data, apps, other users, communities and notebooks. The rich structure of the LabBook metadata graph makes it a good source of information for making these recommendations.

LabBook generates several kinds of context-aware recommendations, from general-purpose recommendations on a user's homepage to personalized content on specific asset pages and personalized ranking of search results. For example, on the user's homepage, recommendations target the user's complete profile, using all related entities from the property graph as context (including the user's social network, datasets she has used, notebooks she has authored, etc.) However, when performing a search in a notebook, browsing another user's profile, or visualizing a dataset, that particular context, as well as the identity of the user, is used to generate recommendations. A search such as "sensors in CA", as illustrated in Figure 6, can leverage relationships between potential dataset candidates and entities in the context (e.g., the issuer, Mary, the search note, "sensors in CA", and the notebook containing it, "Ozone"). For example, the ESRL2 dataset might be preferred over the CA-ARB dataset because Peter is a colleague who has collaborated with Mary while John has not, making Peter's relationship to Mary weigh more than John's.

As illustrated above several factors come into play when making recommendations. These factors can be broadly classified as semantic, temporal, and network (social and otherwise) factors, each leveraging different entities and relationships, and their statistics from the graph. Semantic factors entail features of the graph that represent semantic associations between entities, such as related tables or foreign key relationships. Temporal factors incorporate the time dimension, such as freshness of datasets or recent uses of datasets. Network factors include social connections between people (e.g., followers or co-membership in communities) and distance/paths between entities both in structure (e.g., graph distance) and in semantics (e.g., datasets used in same notebooks or datasets used by a person who is following a community member). For all these factors, statistics are key to understanding the popularity and frequency of access to assets.

While an in-depth description of LabBook's recommenders is beyond the scope of this paper, let's discuss how the recommender system utilizes the metadata graph. LabBook's recommender manager governs a number of entity-specific recommenders, each utilizing a set of features. Features represent some input to the recommender, such as popularity of a dataset, semantic relevance to keywords, etc. Naturally, a dataset recommender considers a different set of features than a person recommender, but some common features might be shared. Each feature is backed by both an offline and an online computation. The offline part performs compute-intensive tasks, such as computing graph statistics and populating the statistics database with information like numbers of followers, numbers of shared notebooks, etc. The online part computes a score, given a specific instance of a potential recommendation, leveraging the graph, statistics, and various indices. The entity recommenders then combine the individual feature scores by applying an entity-specific model that weights each feature to compute an aggregate score.

Recommendations must come with explanations, so that users can understand the rationale behind them and determine whether they are appropriate. Recommenders in LabBook collect explanatory information (such as communities shared between a recommended person and the

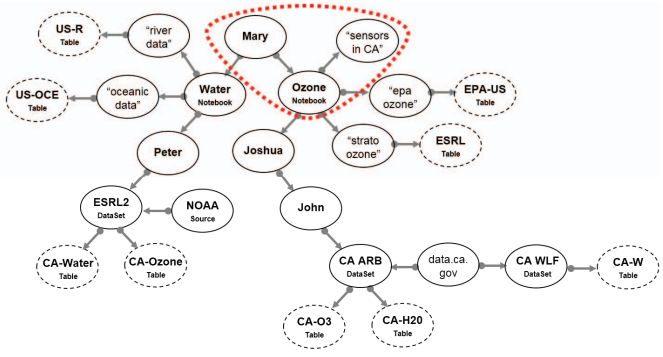

Figure 6. Contextual search leverages recommenders to re-rank results according current context (i.e., person, note, and notebook)

user requesting the recommendation) in their offline processing, and include it as part of the recommended entity during online processing. When recommendations are presented to the user, each recommended entity comes with individual scores for each feature, along with information on how this score differs from the scores of other recommended entities. Icons are placed next to each recommended entity when a particular feature contributed to the entity's score significantly more than others. On demand, users can request further information and see a detailed listing of all associated entities and relationships that contributed to the recommendation. The LabBook recommender manager is extensible in that new features and new entity-recommenders that utilize the graph's monitor, query, and statistics APIs can easily be added to the system.

V. Related Work

A. Metadata Management

As noted in Section IV.A, research in the data management field has historically focused on *schematic* metadata. In a good database design, metadata is created that is both prescriptive (e.g., constraints that ensure the data remains consistent) and descriptive (e.g., unenforced business rules describing an intended semantics and human readable descriptions of column names and values). The main challenge in data integration is to create metadata that relates data from different sources, thus illuminating how they can be combined. Schema mappings, one of the main metadata abstractions used in data integration, are a form of referential constraint that specify how data in one source may be transformed to fit into the structure of another source[15]. When the goal is not integration (representing all data under one structure), but rather alignment (being able to use many data sources collectively), link discovery techniques are often used to identify data in multiple sources that refer to the same real-world entity (entity-resolution problem) [16].

In recent years, more emphasis has been placed on linking schematic metadata to semantic knowledge. Semantic approaches [18] leverage known ontologies to map data attributes to ontological references that serve as the basis for integration. Systems such as DBNotes [19] allow one to attach annotations to data that describe the data or its relationship to external entities, such as people and visualizations. There has been some work on formalizing annotations that record the provenance of data (for example, who created or verified the data).

In general, metadata that records or facilitates collaboration has received less investigation than schematic metadata. Recent work in the data management community aims to close this gap by leveraging people more, from reasoning about provenance, responsibility, and trust [20], to supporting integration incrementally in a pay-as-you-go approach [21], to using crowd-sourcing techniques for querying [22]. Metadata such as data source, context of use, and domain are under-explored, as such information has typically not been captured or represented conveniently in previous work.

Several visual analytic systems have started capturing and using different types of metadata. Provenance is a key aspect of the system design in VisTrails [23]. During exploratory tasks, VisTrails records detailed provenance information, such as input datasets, parameters, and data flow configurations, in a relational database. This information is used for sharing and for simplifying the visualization process, by recommending related visualizations and guiding semi-automated changes to the visualization [24]. While similar in its goals, LabBook is an open web-based data analytics platform, with a diverse set of users, datasets, and analytic tools. It goes beyond VisTrails in capturing and reusing more metadata, particularly collaborative metadata (via social connections between communities of users), contextual metadata (via free-form notes in notebooks and attached resources), and semantic and schematic metadata (via references to ontologies that capture semantics). Such metadata can be helpful to gain a deeper understanding of users' activities.

B. Open Analytics Platforms

Driven by the open science initiatives mentioned in the introduction, a team at the University of Virginia created the Open Science Framework [25], an open platform that enables scientists to manage and share research materials among collaborators, and also increases transparency by making much of the scientific workflow public. OSF is designed partly as a network of research materials, partly as a version control system and partly as collaboration software. It provides a project management system similar to GitHub, but is tailored to support the scientific workflow and management of research materials. It therefore provides specialized features like recording of individual research contributions and project registration for material certification at particular points-in-time (e.g., preregistration for confirmatory analysis). It also provides a flexible add-on system that allows others to contribute to the platform.

Other open analytics platforms include Jupyter [26] and Apache Zeppelin [27], which are collaborative notebook-based interpreters that enable collaborative analysis and visualization for data scientists. These platforms are centralized repositories for researchers to collaborate and track their activities. However, most of the metadata captured is descriptive and individual research materials are not linked to people and tools that could provide context and thereby enable recommendations and further discovery, as is done in LabBook.

C. Open Data and Data Marketplaces

Data is becoming an important commodity. Consumers range from individuals or small organizations to large enterprises, organizations and governments. Multiple sources are curating, publishing, and hosting data. These sources include open data platforms [28][29][30][31], data marketplaces [32], and new emerging platforms such as DataHub [33] and dat-data.com. Open data platforms typically host data of small volume but large variety. The data hosted there is often data published by public or private organizations to fulfill transparency mandates or in support

of citizen engagement. The data is often poorly curated and difficult to use, requiring considerable refinement before it can be of significant value. Data found in data marketplaces is generally of larger volume but lesser variety, and has been curated to a higher degree, as it is for sale. In both cases, however, the discovery process is painful. Users must sift through data catalogs or search for data using specific search terms. To find data that is valuable to them, users must expend considerable effort in a trial and error process, and then figure out how to use it once they have found it. These sources seldom integrate with downstream analytic solutions, and the metadata with which they are annotated is often only skin deep.

More modern alternatives to open data platforms and data marketplaces try to offer a more robust solution, presenting data in a more consumer friendly manner. DataHub for example is a hosted platform for preparing, storing and analyzing datasets, enabling sharing and data reuse. LabBook enables data discovery in a user-friendly manner, by gathering rich collaborative metadata about platform users, the data they interact with and the analytic work they do with that data. Data discovery becomes easier as datasets are recommended to users in a manner that reflects their work, needs, and context.

VI. DISCUSSION

We have argued that open data analysis platforms need to support diverse and evolving analytic ecosystems. Let's revisit the key challenges of such ecosystems: (1) *diversity of people, data, and analytic apps* that make up the ecosystem; and (2) *the dynamic evolving nature of the ecosystem*, where people data and apps and interactions among them change. We discuss how our approach addresses these challenges.

A. Capturing Diverse and Evolving Metadata

Let's first discuss how the metadata graph represents diverse and evolving ecosystems. Here, an important question is whether the property graph is expressive enough to represent such diversity in metadata. Furthermore, what is the right level of granularity for metadata?

Property graphs are attributed, directed, multi-relational graphs. Both nodes and links have an arbitrary number of simple key/value based attributes. As multi-relational graphs, property graphs can have many types of links between nodes. All these features make property graphs powerful for representing general-purpose knowledge, as evidenced by their use in search engines, social networks, intelligent systems, and sciences in general. Property graphs support a flexible schema that can grow easily and be modified on-demand, making them compelling for evolving open data analytics platforms.

We believe property graphs also offer a natural way to represent metadata in open data analytics platforms. In our current implementation we have 10+ entity types and 40+ relationships that represent entities and their corresponding relationships. Naturally, in implementing LabBook we made our own choices regarding the granularity and types of entities and relationships needed to support current use-cases. For example, we don't capture low-level events within apps. We only capture interaction between LabBook and apps as input and output of the applications. However, capturing intra-app, task-level interactions (such as sorting, filtering, and comparing) could open new and interesting directions. For example, we could learn about typical data transformations and could recommend such operations, perhaps even enabling automatic data wrangling, particularly if we correlate such operations with semantic and collaborative metadata.

B. Diversity and Evolution of People, Data, Apps

People come to an open data analysis platform with a wide variety of backgrounds. This makes collaboration challenging because people's information needs and goals vary and so does the trustworthiness of individuals, data, and apps. Simply, different people use different tools and trust in different datasets. Yet, we still would like to be able to help people learn from each other and collaborate with each other. Contextualization is key to supporting such diverse information needs. In LabBook, search results are contextualized, i.e., the person who issued the search and the notebook from where it was issued are important factors in how search results are ranked and presented. We do this as a post-processing step, relating the results to the person and notebook and re-ranking accordingly, weighing different relationships (i.e., paths between the search result entity and the person/notebook) differently. A similar approach is used in recommenders, which employ a number of features that examine the semantic, temporal, and network connectivity between entities to make recommendations of people, datasets, and more. Input to the system is usage, e.g., who used what datasets. As such, if a particular dataset is heavily used in some community, this would be reflected in the ranking of search results for people in that community.

In any meaningful analysis, data from multiple datasets needs to be integrated. For example, in analyzing election data one needs to integrate it with demographic data. In LabBook, we pursued an incremental collaborative approach to developing semantic metadata in support of data integration. For example, tags on a dataset created by one person could be indirectly leveraged by others. More directly, *derivedFrom* relationships capture provenance of past of integrations and could help others find datasets to integrate with. Another aspect of data integration is the multitude of datasets with similar data. This is particularly severe in open systems, where new data is derived and contributed back to the system. While one issue is trust, as we discussed earlier, another is versioning, particularly in evolving open platforms. In LabBook, the metadata graph is flexible enough to allow versioning. For example, references to a dataset could point to the latest release, while records of past uses could point to the version that existed at the time the use occurred.

Lastly, there is also significant variety in the analytic software available. For example, in our metagenomics use case we have nearly 100 analytic apps, which implement various genetic data mining algorithms. Individual bioinformaticians have their own favorite tools for doing analysis, but two critical issues here are interoperability and

provenance. Often work is conducted by using output produced from another app. Using the metadata graph as the central repository and integration point helps in that the graph serves as both a "blackboard system", where apps take in data and put it back and as a "repository system" to capture provenance of actions. While the former helps in composability and abstraction to higher-level tasks, the latter helps people learn from each other, and collectively defines best practices, thus supporting communities of practice.

In the end, the success of open data analytics platforms will be determined by the willingness of the participating communities to openly share data, ideas, processes, and methodologies. Several new efforts are underway to create open science platforms for academic research. Likewise, within large corporations, data scientists are encouraged or even expected to be transparent about their analysis, share knowledge, and collaborate.

VII. Conclusion

In this paper, we described a unified metadata repository that captures information about people, data, tools and their interactions in an open data analytics platform. We argued that such a repository should bring together disparate kinds of metadata, each adding a different perspective, collectively yielding a more accurate representation of how people use data and perform analytic work. Through a specific implementation of such a repository as a property graph in LabBook, we demonstrated several uses of such metadata that could improve the quality of collaboration. We argued that such metadata can be utilized and easily enriched by means of a social user interface with a flexible apps architecture that lowers the cost of contributing metadata, often collecting it as a side-effect of performing analytic work. By applying LabBook in a real-world use-case, we are gaining useful insight on how such metadata could serve a diverse community of users. We believe the rich metadata in LabBook, with a social user experience accelerates discovery through collaboration.

References

[1] P. A. David (2004). Understanding the emergence of 'open science' institutions: Functionalist economics in historical context. Industrial and Corporate Change, 13 (4): 571–589

[2] Z. G. Ives, Z. Yan, N. Zheng, J. Wagenaar, and B. Litt (2015). Looking at everything in context. In Proc. CIDR '15.

[3] Afshinnekoo, E, et al. (2015). Geospatial Resolution of Human and Bacterial Diversity with City-Scale Metagenomics. Cell Systems 1 (1): 72-87.

[4] E. Kandogan, A. Balakrishnan, E. M. Haber, J. S. Pierce (2014). From data to insight: work practices of analysts in the enterprise. IEEE Computer Graphics and Applications '14, 34 (5): 42-50.

[5] S. Kandel et al. (2012). Enterprise Data Analysis and Visualization: An Interview Study, IEEE Trans. Visualization and Computer Graphics '12, 18 (12): 2917–2926.

[6] S. Kandel, J. Heer, C. Plaisant, J. Kennedy, F. van Ham, N. Henry Riche, C. Weaver, B. Lee, D. Brodbeck, P. Buono (2011). Research directions in data wrangling: Visualizations and transformations for usable and credible data. Information Visualization Journal '11,10 (4): 271-288.

[7] Sequencing the Food Supply Chain [Online]. Available: http://www.research.ibm.com/client-programs/foodsafety/index.shtml

[8] Titan Distributed Graph Database. Available: http://thinkaurelius.github.io/titan/

[9] Apache Cassandra database. Available: http://cassandra.apache.org/

[10] Gremlin: a graph traversal language. Available: https://github.com/tinkerpop/gremlin/wiki

[11] Elasticsearch: a distributed search server. Available: https://www.elastic.co/products/elasticsearch

[12] IBM Connections. Available: http://www-03.ibm.com/software/products/en/conn

[13] E. Amitay, D. Carmel, N. Har'El, A. Soffer, N. Golbandi, S. Ofek-Koifman, S. Yogev (2009). Social Search and Discovery Using a Unified Approach. In Proc. ACM HT '09, pp. 199-208.

[14] L. Popa, Y. Velegrakis, M. A. Hernández, R. J. Miller, and R. Fagin. (2002). Translating web data. In Proc. VLDB '02, pp. 598-609.

[15] R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. (2005). Data Exchange: Semantics and Query Answering. Theoretical Computer Science, 336(1):89–124.

[16] O. Hassanzadeh, S. Duan, A. Fokoue, A. Kementsietsidis, K. Srinivas, and M. J. Ward (2011). Helix: Online enterprise data analytics. In Proc. WWW '11, pp. 225–228.

[17] M. S. Fabian, K Gjergji, W Gerhard. (2007). Yago: A core of semantic knowledge unifying Wordnet and Wikipedia. In Proc. WWW '07, pp. 697–706.

[18] L. Getoor and C. P. Diehl (2005). Link mining: a survey. SIGKDD Explor. Newsl. 7 (2): 3-12.

[19] D. Bhagwat , L. Chiticariu , W-C Tan, G Vijayvargiya (2004). An annotation management system for relational databases, In Proc. VLDB '04, pp. 900-911.

[20] Z. Ives, et al. (2008). The ORCHESTRA Collaborative Data Sharing System, In SIGMOD Rec. '08 37 (3): 26-32.

[21] A. D. Sarma, X. Dong, and A. Halevy (2008). Bootstrapping pay-as-you-go data integration systems. In Proc. SIGMOD '08, pp. 861-874.

[22] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin (2011). CrowdDB: answering queries with crowdsourcing. In Proc. SIGMOD '11, pp. 61-72.

[23] S. P. Callahan, et al. (2006). VisTrails: visualization meets data management, In Proc. SIGMOD '06, pp. 745-747.

[24] C. E. Scheidegger, H. T. Vo, D. Koop, J. Freire and C. T. Silva. (2007). Querying and Creating Visualizations by Analogy, IEEE Transactions on Visualization and Computer Graphics, 13 (6): 1560-1567.

[25] Open Science Framework. Available: https://osf.io/

[26] Jupyter (IPython). Available: https://jupyter.org/

[27] Apache Zeppelin. Available: https://zeppelin.incubator.apache.org/

[28] Global Open Data Initiative. Available: http://globalopendatainitiative.org/

[29] Open Data Institute. Available: http://opendatainstitute.org/

[30] CKAN: an open-source data portal platform. Available: http://ckan.org/

[31] Socrata. Available: http://www.socrata.com/

[32] Edd Dumbill. Data Markets compared: a look at data market offerings from four providers. http://radar.oreilly.com/2012/03/data-markets-survey.html

[33] Bhardwaj, Anant, Amol Deshpande, Aaron Elmore, David Karger, Sam Madden, Aditya Parameswaran, Harihar Subramanyam, Eugene Wu, and Rebecca Zhang. Collaborative Data Analytics with DataHub. In Proceedings of the VLDB Endowment, 8 (12):, 2015.