A SANS Database for Machine Learning Models

Christopher Fu and Caitlyn Wolf

March 12, 2018

Motivation

The application of data science is a burgeoning discipline in the field of chemical engineering. There are a multitude of diverse applications where the ability to effectively and efficiently interact with and analyze large, high dimensional data sets would not only be beneficial, but necessary. Specifically, there has been significant advancements in applying machine learning techniques to analyze scattering data, a typically laborious process. [1] Researchers, who actively utilize scattering techniques, frequently travel to national x-ray or neutron source facilities, e.g. NIST Center for Neutron Research (NCNR), to collect this data. However, the continuous data collection results in tens to hundreds of measurements that are difficult to process during the intensive experiment, prohibiting researchers from monitoring or optimizing their work flows on-site. On-the-fly data reduction is a difficult and manual task, but machine learning could assist in on-site decisions by automating these tasks and providing instant feedback to the researchers. To address this difficulty that is encountered by the broader engineering community, we have developed a PostgresSQL database of small-angle neutron scattering (SANS) data that contains experiment conditions, parameters, and raw scattering data.

Due to the breadth of information available in this database, it has the potential to play a significant role in facilitating data processing and help optimize instrument scheduling and maintenance. Specifically, we see this database as a starting point for developing an automated framework for performing data reduction on-the-fly, a necessary step to process raw data into interpretable information. For example, this database would allow researchers to easily query highly specific subsets of the their data. Further, one could perhaps explore previous work in the community making more efficient use of instrument time. Additionally, facility scientists could use this database for access to valuable statistics regarding the use of their instruments, a limited and expensive resource, allowing them to make better decisions for more meaningful instrument use in the future. To date, we have imported SANS data the Pozzo group and publicly available SANS data from NCNR, which we will refer to as "legacy data".

The Dataset

Currently, our database is comprised of 8200 raw SANS data files. While at the heart of a scattering experiment is the 2D raw scattering data (16300 rows x 8 columns per file) gathered at the detector position, there is corresponding meta-data with the instrument, facility, user, experiment, and sample information that is also prudent to include. Measurement run classifications include samples, transmission runs, alignment runs, open beam, blocked beam and empty beam. The different types of measurements are all important for the data reduction process, e.g. background removal or normalization. There are numerous common and unique attributes across these files, such as count time or neutron wavelength, and their implementation into the schema is discussed below (see Schema Creation). Ideally, this data would be uploaded to the database frequently (approximately every 3-6 hours), but due to the large size of this data we expect this to be a significant consideration during not only implementation, but in subsequent iterations of this schema design as well. Facility-specific procedures were used to extract this information from the binary raw data files provided for integration into our PostgresSQL database, and at this stage, we have manually linked specific measurement relations, e.g. sample run with its corresponding transmission file, for the Pozzo group data subset. In total we have loaded approximately 15 GB into our database, but expect this total to rapidly grow as we begin importing more legacy data (i.e., one decade of data for instrument corresponds to about 150 GB). While we had to rely on manually integrating the data into our database, we anticipate that the logical next steps of the project, beyond this quarter, are to migrate the database to a cloud based service, such as AWS (Redshift), and to develop an automated process for reading the meta-data from SANS data files.

Related Work

We are not the first to explore the value in a scattering database serving the community. There are a few publicly available scattering databases available, but these systems have a limited range for specific areas of research and also rely on researchers submitting their published data to the database. [2-3] The most similar database to ours is called SASBDB [2], which uses MySQL to host the database. They have a somewhat similar schema in that they separate their experiment parameters and their data into separate tables, which are also joined to a unique user ID. The main deviation from the SASBDB and ours is that our database is constructed with the intention to host all of the raw data for files corresponding to a given experiment so it can be functional for processing, where as SASBDB hosts published results that have already been processed and polished. To put another way, our intention is to host raw data to help researchers query and work with their data, where as SASBDB is to provide a reference to published data.

While these systems provide value to the subsections of the scattering community they serve, we believe that our project encompasses a much broader, and far more noble, impact as previously discussed. All data collected at the NCNR on their neutron scattering instruments is made publicly available [4], but it currently does not exist in an easily accessible manner for researchers to query from the dataset as a whole. Our project focuses on build-

ing on this existing structure to make a database system robust enough to implement this comprehensive dataset while allowing scattering researchers and facility scientists to make powerful, meaningful, specific queries across multiple research fields.

Results

Schema Creation - Serving Two Masters

A focal point of this project was designing a suitable schema for this database. One of the unique challenges with this dataset is to create an infrastructure that can serve different parties, from system facility administrators to the broader community of scattering researchers who would want access to the experiment information. The overall schema of our database is presented below in Figure 1, including previously discussed attributes regarding the experiment, instrument, facility and user for each measurement. While some tables are somewhat intuitive, such as users and instruments, many of the tables are created to distinguish between the different types of measurement runs that are conducted in a given experiment (note in this context experiment corresponds to a collection of trials conducted for a given grant or trip). Each trial conducted has an associated data file with at least eight of the attributes shown in Figure 1, which is also associated with a given instrument. Further, an individual data file can be cast as a sample run which is used to collect SANS data, a transmission run to recover baseline information, or a beam characterization run in the form of alignment, empty beam, open beam, or blocked beam conditions. Because the transmission runs are required to obtain baseline readings that are used in sample data reduction, we provide a join table to the sample runs.

Our schema has undergone one significant revision since our milestone, where we have added one more table called "RawData" which contains the raw scattering data corresponding to each file and its meta-data. The raw data corresponds to a 2D X-Y grid, with 6 measurements for each point on the grid. While including all of the raw data may seem ambitious, the ability to query this data will be valuable in future applications of this database. Because many entries in this table will correspond to the same data file, we had to use a composite key as a primary key, comprised of the filename, with a specific X-Y coordinate.

Gaining Insights Through Queries

As noted above, our database is comprised of data accumulated in the Pozzo group and some legacy data acquired from NCNR. Having this cornucopia of scattering data has allowed us to carry out queries from the perspective of different database users: a researcher hoping to analyze his/her own data and a facility scientist charged with scheduling and maintaining the instruments. An example of such information is shown in Figure 2, which illustrates summary statistics of typical experiment run times. The results shown in Figure 2 are enlightening for those planning a scattering experiment, as a key step in trip planning is submitting a detailed proposal requesting a specific amount of instrument time. Having this break down, perhaps one even further refined for each user, can allow researchers to produce more accurate estimates for instrument time, not only strengthening their research plan, but

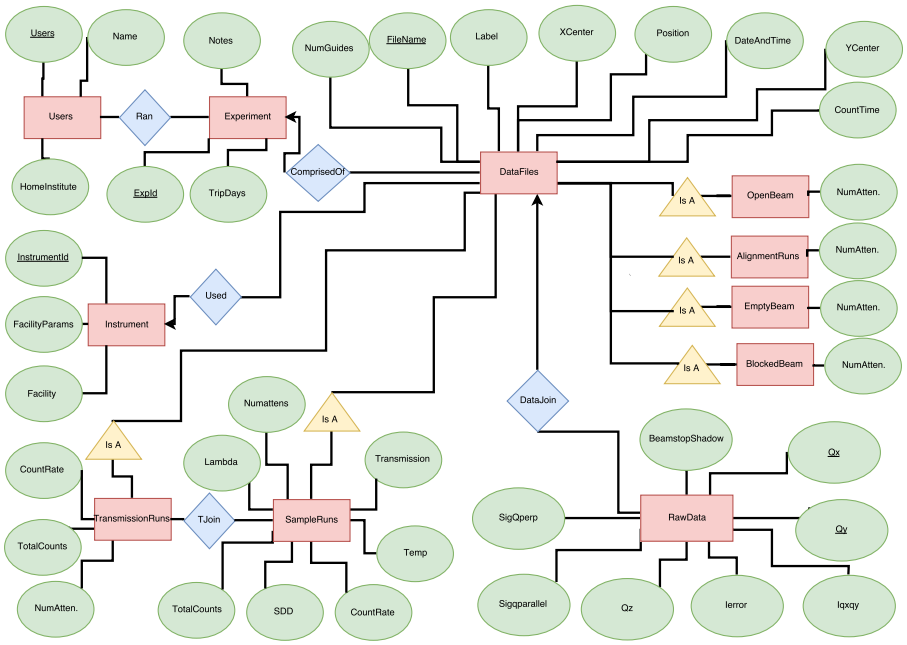

Figure 1: Schema of the constructed database. Tables are shown as red blocks, attributes are green circles, blue diamonds represent join tables, and yellow triangles indicate "Is a" relationships. Primary keys for each table are underlined.

also increasing the likelihood that their proposal is approved.

Additionally, this database can be of use to facility scientists. For example, because these instruments are in high demand year-round, beam time is tightly scheduled and allocated to specific users. As a result, keeping track of instrument usage and scheduling maintenance time must be done in an effective manner to keep instruments operational with minimal interruption (see Figure 3). Also, having a characterization of how instruments are being used, and potentially by which users, can allow these facility scientists to be better equipped to offer assistance or troubleshoot problems (see Figure 3).

Evaluating Performance

While the primary objective of this project was to design and implement a functional schema for storing and querying raw SANS data, it is prudent to evaluate how well this database performs in terms of running queries. Using the EXPLAIN ANALYZE function in PostgreSQL, we recorded the timings for the various queries we conducted (shown in Table 1). It is clear from Table 1 that obtaining summary statistics about an instrument's use can be recovered fairly quickly. This is not completely surprising as these queries do not interact with the raw scattering data. Instead, it utilizes a fairly small data set as the DataFiles table is only approximately 8400 rows x 8 columns) joined to much smaller tables for these queries.

The most disheartening result however comes in when we want to query information

[1]standard deviations shown in parentheses

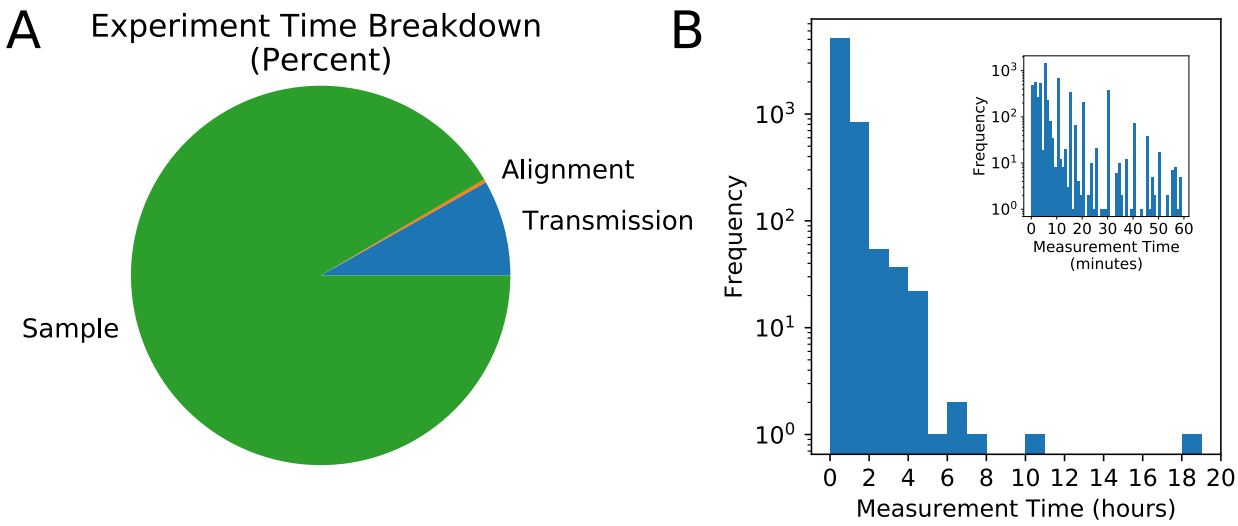

Figure 2: A) Average break down in terms of percent for time spent carrying out alignment, transmission, and sample runs. B) Histogram showing the frequency of measurement times for data collection. The inlet shows a break down for runs under one hour.

Table 1: Timing Information for Executing Queries Without Raw Data

Query Recovering Objective	Average Query Execution Time (ms)[1]
Experiment Time Breakdown	30.6 (1.3)
Ave. Trip Time per Exp.	19.6 (0.6)
Frequency of Run Times	22.1 (0.6)
Total Run Time per Month	15.3 (0.17)
Number of Measurements per Wavelength	2.9 (0.1)

regarding the raw scattering data. To evaluate this, we ran a set of queries to recover the maximum intensity value for a given data file, but adjusted the number of files we wanted to recover the information for (i.e., we specified filenames). Figure 4 indicates the run times for these queries as a function of the number of files we specified. It is clear that the query run time grows linearly with the number of files specified, which is very unappealing if we wanted to obtain statistics for all of the raw data files in our database. Assuming the trend in Figure 4 holds, obtaining the maximum intensity for every data file would require approximately 4.5 hours. Note that this run time does not require any sophisticated SQL operations, but is merely a filter. Therefore, it is clear that while serviceable for this quarter-long project, a PostgresSQL database is not suitable for efficient querying of the data.

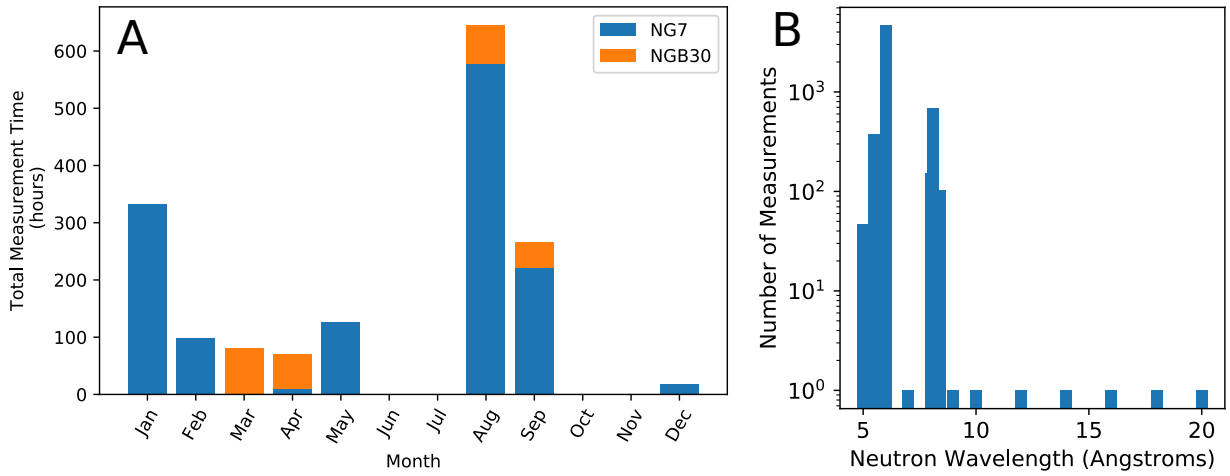

Figure 3: A) Breakdown of total run times for different instruments by month B) Histogram of neutron wavelengths used in experiments.

Challenges & Future Work

A challenge that was not anticipated was how unwieldy the 2D scattering data files became. The data files are stored as binary text files that at this point needed to be manually converted to usable information via an external program. In addition, because we put a premium on storing raw data, allowing users to apply and filter as they saw fit, this meant we had to store a lot of extra features for the raw data files. Rather than storing a single intensity value per grid point in a raw data file, we were also forced to incorporate other attributes (see Figure 1) which increased the size of the dataset. Currently there lies a distinct conflict with our goal of allowing users to interact with and query raw data, and facilitating these queries in an efficient manner. Because this problem will only compounded as we increase the database size, the next step of this project is the migrate our database to a commercial cloud service, such as Amazon Web Services.

Conclusions

In this project we have designed and constructed a database for raw SANS data files. The versatility of the schema not only allows for instrument users to query their raw scattering data for analysis, but also allows for experimenters and facility managers alike to query summary statistics about how instruments are used. While we were able to establish a "toy-database system", it is clear that a PostgresSQL database is not sustainable for the future as the size of the database continues to grow. Moving forward we will begin investigating commercial, cloud-based options, such as Redshift (AWS). However, this project demonstrates an important first step in establishing a sustainable, comprehensive SANS database that can serve and support the broader engineering community.

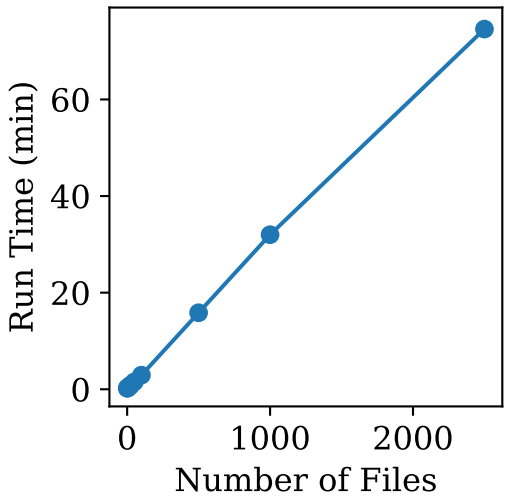

Figure 4: Query execution time in minutes as a function for the number of files specified in the query to recover maximum intensity value in file.

Breakdown

The authors contributed equally to this work.

Acknowledgements

We acknowledge the contributions from the Pozzo Research Group, specifically Dr. Lilo Pozzo and graduate student Yuyin Xi, in providing a valuable SANS dataset and meaningful discussions for this work. The authors also wish to recognize data contributions and valuable insight from Paul Butler, a SANS scientist and team leader at the NCNR. This work was facilitated though the use of advanced computational, storage, and networking infrastructure provided by the Hyak supercomputer system at the University of Washington.

References

[1] Yoon, C. H. et al. Unsupervised classification of single-particle X-ray diffraction snapshots by spectral clustering. Opt. Express 19, 16542 (2011).
[2] Valentini, E., Kikhney, A. G., Previtali, G., Jeffries, C. M. & Svergun, D. I. SASBDB, a repository for biological small-angle scattering data. Nucleic Acids Res. 43, D357–D363 (2015).
[3] Kikhney, A. G., Panjkovich, A., Sokolova, A. V. & Svergun, D. I. DARA: a web server for rapid search of structural neighbours using solution small angle X-ray scattering data. Bioinformatics 32, 616–618 (2016).
[4] NCNR Neutron Data Access Policy. (2010). Available at: https://www.ncnr.nist.gov/news/data_access_policy.html.