IMEKO TC6 International Conference on Metrology and Digital Transformation
September 19 – September 21, 2022, Berlin, Germany

TOOLS FOR A SEARCHABLE AND TRACEABLE CURATED DATABASE

Frederic Brochu [a], *Michael Chrubasik* [a], *Spencer A. Thomas* [a, *]

[a] Data Science (National Physical Laboratory), Teddington, UK
* Corresponding author. E-mail address: spencer.thomas@npl.co.uk

Abstract – We present a framework for easy annotating, archiving, retrieving and searching measurement data from a large-scale data archival system. Our tool extends and simplifies the interaction with the database and is implemented in popular scientific applications used for data analysis, namely MATLAB and python. This allows scientists to execute complex interactions with the database for data curation and retrieval tasks in a few simple lines of accessible templated code. Scientists can now ensure their measurement data is well curated and FAIR (findable, accessible, interoperable and reusable) compliant without requiring specific data skills or knowledge. Our tools allow users to perform SQL type queries on the data from simple templated scripts allowing data retrieval from long term storage systems.

Keywords: searchable metadata, reproducibility, data curation, data traceability, FAIR

1. INTRODUCTION

Technological and scientific advances over the last 20+ years have led to the ability to generate and store vast amounts of data. Furthermore, emphasis on reducing acquisition times has significantly increased the throughput of data from experiments. In parallel with the developments in measurement technologies, there have been significant advancements in data storage, allowing this data to be efficiently captured and stored. However, a lack of systems or standards to organise or curate data leads to ad-hoc file structures, inconsistent conventions in recording metadata and a loss of data provenance. Consequently, the vast amounts of data being recorded are often not findable, accessible, interoperable, and reusable (FAIR) [1].

Without a well curated database [2] that includes rich metadata, data will not be findable. For research institutions that generate or collect large volumes of data this is highly problematic as it significantly restricts the data's reusability and therefore value. This is compounded with potentially high financial costs associated with acquisition and storage if the data are not findable. The inability to retrieve data may have significant repercussions for reproducibility of results, traceability, or adhering to funder requirements. The US National Institutes of Health (NIH) have now issued a mandate for all their funded research include data management and sharing plans [3]. The policy that *science data* needed to "validate and replicate research findings, regard less of whether the data are used to support scholarly publications" must be shared by researchers. It is estimated that tens of billions of US dollars of research funding lack of clear and traceable experimental workflows for independent validation [4] and 10-50 billion US dollars funds research that uses irreproducible or 'deficient methods' [3].

The concept of *measurement traceability*, where any instrument's measurement can be linked to a known standard though an unbroken chain of comparisons, is well established. *Data traceability* extends this concept to data and analysis pipelines where a given output (processed data, figures, statistical tests, etc) is linked back to data at the point of measurement through an unbroken chain of steps in a data workflow. These steps include data conversion, data processing (for example noise reduction) and data analysis steps (e.g. statistical tests, machine learning, etc) [2]. Throughout the rest of this paper, we use the term traceability to refer to *data traceability*.

Previously, at the National Physical Laboratory (NPL), we have developed methods for curating data at the point of measurement which can be used to establish a FAIR and traceable database [2,5]. A curated database with relevant well-structured metadata tags permits searches using structured query language (SQL) or similar, where investigators can return a list of datasets within the database that match a given set of criteria. The metadata itself can be analysed to reveal insights into the data. For example, in radiology, analysis of the sensitivity and radiation exposure over time at different sites uncovered inter site and temporal differences [6]. However, to establish such a system requires a high level of computational skills and often requires bespoke software creating significant barriers for measurement scientists.

We use our internal database for long-term curated storage of measurement data along with experimental conditions that form the basis of the metadata and are vital for traceability and reproducibility. We introduce a tool that combines and extends the functionalities of the application program interface (API) provided with NPL's archive for file transfer, annotation, and metadata queries into a single, convenient interface tool accessible from popular scientific applications MATLAB and python. This tool not only simplifies interaction with the archive, also called the "Objectstore" in the following, but it also makes the entire data management process more accessible for scientists. Moreover, this tool allows investigators to easily retrieve any data stored and provide direct access to all relevant data, metadata, codes and the complete processing pipeline. Data can then be retrieved from long term storage and remain fully reproducible with traceable data workflows for auditing or sharing purposes.

2. METHOD

The developed tool is a set of MATLAB and python scripts handling interactions with the archive data storage and experimental data with metadata encapsulation. This enables “behind the scenes” operations at the command of the scientists wishing to archive their data without requiring the programming skills necessary to do so. The interface tool is invoked through MATLAB which is used as a user interface for python code operating as a two layer program. The first layer is a MATLAB master class describing a connection “object” with different call-back functions. In the second layer the functions are mapped into a python layer handling “representational state transfer” (REST) calls [7] to the Objectstore APIs for file handling and metadata queries. File transfers are performed with the AWS S3 protocol [7]. The layout tools and their different components are presented in **Fehler! Verweisquelle konnte nicht gefunden werden.**.

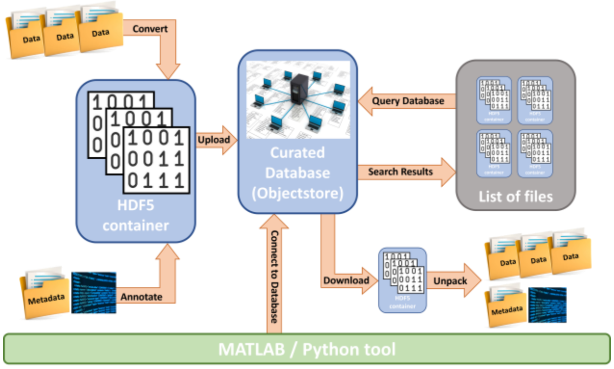

Fig. 1. Interface layout and functionality described in section 2. User functions are represented by orange arrows.

2.1. HDF5 container

We store the data in a Hierarchical Data Format (HDF5) container file that can hold an arbitrary number of data files and formats. This allows the storage of experimental data with associated datasets such as calibration data or processing scripts as a complete and unbroken data pipeline [5]. Containerising the data and associated steps in a workflow ensures reproducibility though the data pipeline. The data parsed into the container are stored in binary form as this allows support of any data format. They are also compressed at the creation of the container to optimise data replication to the archive system.

2.2. Annotation

The metadata for complex experiments is multi-source [8] and is collated into a single well-formed XML [9] file outlined in [5]. This XML file is used to tag the HDF5 container with the associated metadata by assigning the attributes of the HDF5 file with the definitions and values from the XML file. Linking the HDF5 with its associated metadata ensures the data are FAIR compliant prior to uploading the file to the database. Although aiding reproducibility and interoperability for individual files, linking the data with metadata is not sufficient to provide a curated database that can be easily searched or mined for meta-analysis. By using standardised and well-formed metadata to link to the HDF5 files we can automatically establish a curated database. That is, all HDF5 containers have the same metadata structure and are therefore well organised and can be viewed and search in a systematic way. Further details of this are in Section 2.4.

2.3. Objectstore

The annotated HDF5 container files are stored in our “Objectstore” database. The Objectstore is NPL’s large scale data archival system, an instance of the Hitachi Content Platform (HCP), which in its most basic form is a flexible database for storing and annotating data. File annotations tag the data with associated metadata for curation and database searching. The metadata can be defined by the user, known as ‘custom metadata’, and can be used for curating complex experimental data [2]. When tagged with metadata, data are stored in an internal database with a dedicated API allowing simple SQL type queries for searching the data.

The Objectstore consists of “Tenants”, organisational level division of the system (e.g. departments); and “namespaces”, logical grouping of objects (e.g. projects). This archival system supports file versioning as well as annotation. A database with versioning enabled can enhance traceability when the data are tagged with the associated metadata [5]. Both the data and metadata can be updated for new versions.

The tool provides specific functions for: connecting to the database, uploading data following archiving as a HDF5 container file [2], downloading datasets, and performing metadata queries (see Section 3 for more details). An example of the MATLAB script to establish the database connection is given in **Fehler! Verweisquelle konnte nicht gefunden werden.**.

```
% create a connection object
conn = NPLObjectStore_MvP()
% log in with user credentials
% …
% connect to a namespace (specific
collection of objects e.g. a project)
within a Tenant (division of the
Objectstore e.g. organisational
departments)
conn.full_setup('namespace','Tenant')
```

Fig. 2. MATLAB script to setup database connection. Comments are given in lines beginning with % and coloured green.

Once a connection is established the data can be uploaded simply by specifying the location of the data to be uploaded (`local_dir`) and the target storage location on the database (`database_dir`) in a function call. The function first creates a HDF5 container file that is populated with the data in the specified directory, tags the container with the metadata as [5] before uploading the contained to the database. The script for this is given in **Fehler! Verweisquelle konnte nicht gefunden werden.**.

```
% create a HDF5 file, tag with metadata
then upload to the database
local_dir = 'C:\path\to\data'
```

```
database_dir = 'DBdir1/DBdir2/'
[status, output] =
conn.archive_and_upload(local_dir,
database_dir)
```

Fig. 3. MATLAB script to upload local data to the database. Comments are given in lines beginning with % and coloured green.

Similarly, data can be easily downloaded from the database. In this case `local_dir` is the where the HDF5 container will be downloaded to and then unpacked to the same folder structure and data formats as the originally uploaded data. Directories that were originally shortcuts are unpacked as subdirectories within the data parent directory, e.g. there are no shortcuts when data are downloaded and unpacked. The script for downloading the data is given in **Fehler! Verweisquelle konnte nicht gefunden werden.**.

```
% download and unpack a HDF5 file
database_dir =
'st3/TracebleTest/TracebleData.hdf5';
local_dir =
'C:\Users\st3\from_Objectstore';
[status, output] =
conn.download_and_unpack(data_folder,
objectstore_area)
```

Fig. 4. MATLAB script for downloading data from the database to the local computer. Comments are given in lines beginning with % and coloured green.

2.4. List contents and SQL queries

The curated database consists of HDF5 files tagged with associated metadata that allows the entire database to be searchable. The content of the entire namespace, or a specific directory can be listed as shown in Fig. 5.

```
% a list of all folders in namespace
conn.list_objectstore_directory(' ')
% list files in dir1/dir2
conn.list_objectstore_directory('dir1/di
r2')
```

Fig. 5. MATLAB script to list the objects contained within the Objectstore namespace. Comments are given in lines beginning with % and coloured green.

We can also search the content of our database filtering on any of the tagged metadata attributes using SQL type queries. The queries return a list of HDF5 files that match the criteria of the query. In addition to attributes of the data from the metadata, the queries can also include the filename or a unique identifier making all the data findable. Using SQL type queries, which can be standardised through template queries for metadata attributes in a specific domain (see Results Section), ensure that the data are accessible to all users in line with the GO FAIR principles [10]. Database permissions can be set to restrict the visibility of data for different users as required, though this is outside the scope of this work.

3. RESULTS

We demonstrate our tool using an exemplar case study using a database from the Cancer Research UK Rosetta Grand Challenge (A24034) Project led by NPL's NiCE-MSI group [11]. Previously we have established a curated database of complex experimental data [2] that can be used for traceable data processing workflows that are fully reproducible [5]. This database consists of large volumes of experimental data acquired from several different instruments (vendors and models), across multiple sites, with a large number of experimental parameters and operating procedures that are dependent on the sample. The project also conducts cohort, longitudinal and inter-laboratory studies, and aims to conduct some meta-analysis on the data once complete. With the database connection opened as shown in Figure 2, the database can be searched with simple SQL queries of the form `conn.metadata_query('key'  ,'value')`. This provides a simple user-friendly means for experimentalists to search the curated database for any subset of files making the data findable. Some instruments write data in a proprietary format that is accessible internally, but this access is lost when archiving the data as the proprietary format requires the instrument's vendor software to open. We convert this data in an open community standard format called imzML [12] making the data accessible. Some examples of domain specific queries that users may want to perform are:

- data from a specific measurement technique, for example DESI and MALDI
  `conn.metadata_query('Technique',`
  `'DESI')`
  `conn.metadata_query('Technique',`
  `'MALDI')`
- data from a particular experimental study
  `conn.metadata_query('Study',`
  `'SLC7A5')`
- data acquired from samples from a particular collaborator `conn.metadata_query`
  `('SampleSource','AstraZeneca')`
- data from a specific vendor instrument model, for example a 'SYNAPT G2-Si' model
  `conn.metadata_query('Instrument',`
  `'SYNAPTG2-Si')`
- data from a particular sample (unique barcode from a separate sample management database, the data is integrated in to our curated database prior to upload see [2])
  `conn.metadata_query('BARCODE',`
  `'1000202')`
- datasets of the same experimental parameters eg, data of a particular instrument polarity
  `conn.metadata_query('Polarity',`
  `'Negative')`
  size of acquisitions area for each pixel
  `conn.metadata_query('PixelSize',`
  `'100 microns')`
  acquisition time for each pixel
  `conn.metadata_query('ScanTime',`
  `'0.485 sec')`
  data acquired over the same measurement range
  `conn.metadata_query('massRange',`
  `'m/z 50-1200')`

Note the last three examples can be executed with or without the units.

The queries return a list of matching file ID's and version ID allowing users to select these for download or perform meta-analysis on the results. A representation of the results for the `conn.metadata_query( 'massRange', 'm/z 50-1200')` query is given in Table 1 where additional fields are provided and categorised to highlight the contents of the HDF5 objects to a general audience. Each of the objects can be downloaded and *unpacked* back to the original format of the data and metadata with all the associated experimental conditions.

Table 1. Example output from the query conn.metadata_query('massRange', 'm/z 50-1200') with some additional fields with categorised results for clarity of the general reader

ID	Version	Site	Modality	Sample
Object A	1st	A	B	A
Object B	2nd	A	A	A
Object C	1st	C	A	B
Object D	4th	B	C	B
Object E	2nd	C	C	B
…				

This search functionally ensures all the data in the Objectstore are findable and accessible. As the data is stored in HDF5 files with well-structured metadata, they are interoperable and re-useable. Our tool provides a simple interface for measurement scientists to establish and interact with a curated database without requiring bespoke computational skills. This ensures scientists are adhering to good practices in data handling and storage and have direct access to all the experimental conditions and settings. The organisation and curation of data is a time-consuming task and increases in difficulty with time since carrying out the measurement. As the majority of institutions and laboratories do not have designated data managers [3] this tool offers a practical solution for this problem.

4. CONCLUSIONS

We have introduced and demonstrated our tool for non-experts to interact with a well-curated database via popular languages MATLAB and python. Encapsulating several complex APIs into a single and user-friendly tool allows measurement scientists to easily ensure that their data is stored in a well curated, FAIR compliant and traceable database without the need for advanced computational skills. The ability to collate all relevant data, link with relevant machine actionable metadata, and upload to a database with a single function call reduces the computational barrier significantly. Linking data with standardised and well-structured metadata established a curated database automatically when data is uploaded. This curated database provides capabilities for efficient data search and retrieval, as well as the possibility of meta-analysis.

A single line of code is used to perform SQL type searches of the database ensuring all data are findable and accessible. An integrated function for downloading and extracting the data in its original format allows the utilisation of the HDF5 container without requiring the user to interact with it. Storing the data linked with its associated metadata alongside codes and programs ensure interoperable and reusable data that is also traceable. We demonstrate this through an exemplar case study of data collected from a large-scale multisite imaging project with large volumes of highly complex measurement data.

Template scripts further reduce this barrier and enable metadata capture at the point of measurement as well as any stage throughout the data processing pipeline. This enables experiments to easily retrieve data and maximise the usefulness of a FAIR and curated database without requiring any knowledge of these principles.

ACKNOWLEDGMENTS

This work was funded by the Department for Business, Energy & Industrial Strategy through the UK's National Measurement System. The authors would like to thank Paul Duncan (NPL) for helpful comments and feedback on the manuscript.

REFERENCES

[1] M. D. Wilkinson et al, *The FAIR Guiding Principles for scientific data management and stewardship*, Scientific Data 6(1) 2019

[2] S. A. Thomas and F. Brochu, *A framework for traceable storage and curation of measurement data*, Measurement: Sensors, 18(2021):100201 2021

[3] Max Kozlov, "NIH ISSUES A SEISMIC MANDATE: SHARE DATA PUBLICLY ", Nature Vol 602:558-559, 24 February 2022.

[4] L. P. Freedman, I. M. Cockburn, and T. S. Simcoe, "The Economics of Reproducibility in Preclinical Research," PLOS Biology, vol. 13, no. 6, p. e1002165, Jun. 2015, publisher: Public Library of Science.

[5] S. A. Thomas and F. Brochu, *Curation at the point of measurement and traceability of measurement workflows*, Measurement: Sensors, *(in review)*, 2022

[6] Santos, M., Sá-Couto, P.; Silva, A., Rocha, N., "DICOM metadata-mining in PACS for computed radiography X-Ray exposure analysis: a mammography multisite study", European Congress of Radiology-ECR 2014: 2014.

[7] AWS S3 REST API protocol, AWS, https://docs.aws.amazon.com/AmazonS3/latest/API/s3-api.pdf#Welcome (accessed 21 February 2022)

[8] N. A. S. Smith, Nadia, D. Sinden, S. A. Thomas, M. Romanchikova, J. E. Talbott and M. Adeogun, *Building confidence in digital health through metrology*, The British Journal of Radiology, 93(1109) 2020

[9] “Well-formed-XML”, W3 resources, https://www.w3resource.com/xml/well-formed.php (accessed 21 February 2022)

[10] GO FAIR, FAIR Principles https://www.go-fair.org/fair-principles/ (accessed 21 February 2022)

[11] Rosetta Project, Cancer Research UK Grand Challenge https://cancergrandchallenges.org/teams/rosetta

[12] Andreas R¨ompp et al. imzML: Imaging mass spectrometry markup language: A common data format for mass spectrometry imaging. Methods Mol Biol., 696:205–224, 2011