



A tool for curating and searching databases proving traceable analysis of data and workflows

Frederic Brochu¹, Michael Chrubasik¹, Spencer A. Thomas¹

¹ Data Science, National Physical Laboratory, Hampton Road, Teddington, Middlesex, TW11 0LW, United Kingdom

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 (Structured Query Language) queries on the data from simple templated scripts allowing data retrieval from long-term storage systems.

Section: RESEARCH PAPER

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

Citation: Frederic Brochu, Michael Chrubasik, Spencer A. Thomas, A tool for curating and searching databases proving traceable analysis of data and workflows, Acta IMEKO, vol. 12, no. 1, article 12, March 2023, identifier: IMEKO-ACTA-12 (2023)-01-12

Section Editor: Daniel Hutzschenreuter, PTB, Germany

Received November 18, 2022; **In final form** February 16, 2023; **Published** March 2023

Copyright: This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Funding: This work was funded by the Department for Business, Energy & Industrial Strategy through the UK's National Measurement System.

Corresponding author: Frederic Brochu, e-mail: frederic.brochu@npl.co.uk

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 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 repeated acquisitions and double storage if the data are not discoverable. The inability to retrieve data may have significant repercussions for reproducibility of

results, traceability, or adherence to funder requirements. The concept of measurement traceability, where any instrument's measurement can be linked to a known standard through 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], [3]. 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 to match given 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 [4]. However, establishing 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. In this work, we introduce a tool that combines and extends the functionalities of the application programming interface (API) provided with NPL's archive for file transfer, annotation, and metadata queries into a single, convenient interface tool accessible from the popular scientific applications MATLAB and Python. This tool not only simplifies interactions with the archive, but it also makes the entire data management process more accessible for scientists. The data archive we use in this work is an 'Objectstore', a database storing data as objects that have their own attributes comprising of system metadata (size, creation date, etc) and custom metadata (user-definable fields). In this work we will exploit the archives' use of data objects to store any number of multimodal data files, in any format, as a single Hierarchical Data Format (HDF5) file that we use as a 'container' for data files. A single HDF5 container file, consisting of any number of data files, corresponds to the data objects uploaded to the data archive. We refer to the data objects as HDF5 container files through the manuscript. We can further exploit the Objectstore functionality by utilising the custom metadata to define domain-specific attributes that are used to 'tag' the HDF5 file in the Objectstore enabling highly specialised and domain-specific searching of the data. For large organisations this provides a flexible approach to automatically curate large and diverse databases without domain-specific infrastructure. For users, this enables multiple data files to be wrapped in a single container that is stored in a data archive for long-term storage alongside relevant metadata for searching and retrieval. Our tool provides a user-friendly interface for users to perform all necessary steps (generating the container, tagging with metadata, uploading to the database, performing searches and retrieval of the data) without any expertise in these areas. We provide a case study using large and complex multimodal cohort data from experiments conducted at multiple institutions and across multiple instruments.

2. BACKGROUND

Data curation is the process of organised storing of data in a structured way with rich machine-actionable information about the data and its provenance. Analogous to finding a book in a library, data curation enables users to locate specific datasets based on a defined list of attributes. For example, locating a dataset consisting of an image of a *cat* in *winter* captured with a *mobile phone camera* in the *countryside*. Although many databases will enable the searching of the criteria of 'images', 'cat', 'camera phone', 'winter', and 'countryside', there is no strict matching of attributes that these keywords map to. In a curated database we can perform such searches, type = 'images', subject = 'cat', device = 'camera phone', season = 'winter', and location = 'countryside'. This search would provide exact matches to our search rather than any attribute matches to the keywords in the former case.

The demand for long-term data curation arises from the researchers themselves, as it enables them to utilise the data in future studies, from funding bodies through data retention requirements, and from the community, which promotes open science with FAIR data. Well-curated data has the additional

benefit of enabling meta-analysis across the database which can provide more precise estimates compared to individual studies, as well as an assessment of variability [5] and the development of computational models [6]. For example, meta-analysis has identified a higher proportion of positive COVID-19 tests in low and low-middle income countries compared to higher income countries [7], evaluating the treatment effects [8] and by assessing the impact of missing data on outcomes [9].

There is currently no data curation tool or platform suite to manage experimental data due to its inherent complexities. This is particularly problematic for research which will typically be subject to funding bodies' data storage and retention policies. Due to the high cost (financial, expertise, and time) involved in some experimental studies, researchers want to maximise the future utility of the data in other studies. For example, healthcare or pharmaceutical studies involving tissue imaging have very complex data collection pipelines with different centres responsible for collecting the samples (e.g. biopsies), sample preparation (e.g. embedding and sectioning) and the measurement data (e.g. imaging). In this case, one experiment can involve multiple institutions and the provenance of the sample is highly complex. Data quality controls and future meta-analysis require this information to be captured in a machine-actionable way.

Current solutions range from individual-level record keeping to universal data repositories; we argue these are insufficient for curating measurement data. Record keeping, such as spreadsheets or a database (SQL, Access, etc), to capture information, such as data storage location, does not constitute a data archive as it simply lists the locations and possibly some metadata. The fact that this information is unstandardized, prone to error, not machine-actionable, and not searchable (in a database environment) is even more problematic as it prevents the data from being FAIR. Universal data repositories such as Zenodo, Figshare, Scientific Data, Dropbox, re3data, etc, offer storage of data and user flexibility with regards to files and formats stored as well as providing some scope for metadata. However, capturing the metadata is far from trivial [2], [3], [10] and there may be many terms to include. Although many of these platforms offer a search functionality, it is offered in a basic implementation, preventing highly specialised searches such as structured database queries. Furthermore, there are no specific checks on these entered fields, hence information may be missed, incorrectly added, or exist in several forms (e.g., acronyms, capitalisation).

3. 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 [11] to the Objectstore APIs for file handling and metadata queries. This configuration not only provides access of our tool for MATLAB and Python users, but also provides a simple interface (in MATLAB) for users with little programming

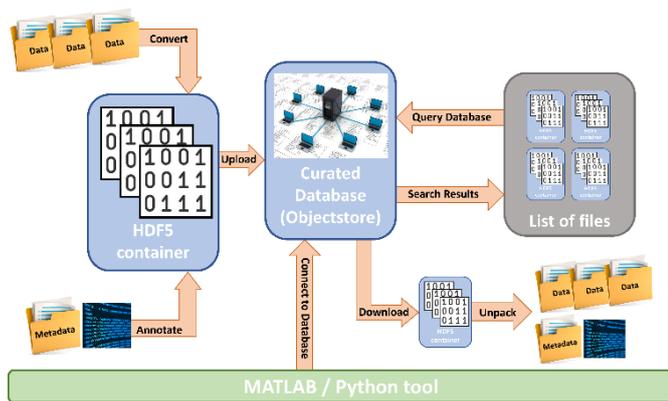


Figure 1. Interface layout and functionality described in section 3. User functions are represented by orange arrows. The desired datasets are each converted to binary and added to a HDF5 container file that is annotated (or tagged) with relevant metadata. The HDF5 container file is uploaded to the curated database where the metadata can be queried. Any desired data can be easily downloaded and automatically converted back to its native format.

expertise. The python libraries are installed as dependencies of the code following our documented installation procedure, this also covers MATLAB installation which is straightforward. This tool is designed to work within an organisations digital infrastructure and thus can be preloaded on to institutional machines such as laptops and lab machines. File transfers are performed with the AWS S3 protocol [11] with all data stored on our institutions internal data storage infrastructure with access permissions fully controllable by our IT team. Our code is available on a private GitLab repository as a PyPI package, and a public release may be possible in the future.

To be an effective solution, our tool has required: to store all relevant and related data sets as a single instance (see Section 3.1), the data be linked to the metadata (see Section 3.2), all data to be archived in a common location (see Section 3.3), and the metadata be searchable for data retrieval (see Section 3.4). The layout tools and their different components are presented in Figure 1.

3.1. 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 [3]. Containerising the data and associated steps in a workflow ensures reproducibility through the data pipeline. The data parsed into the container are stored in binary form as this allows supporting any data format. They are also compressed at the creation of the container to optimise data replication to the archive system.

3.2. Annotation

Annotation refers to the process of attributing information to a data file such that it becomes the metadata for that file and providing terms that the file can be searched with. The metadata for complex experiments is multi-source [10] and takes different format. We collect and aggregate them all in a single ‘well-formed’ XML file [12]. XML is a metadata format that is both human and machine-readable, and is the only format supported by the NPL archive described in Section 3.3. The metadata flow is duplicated: a copy is embedded in the HDF5 container, and another is used to link the HDF5 container with the associated

```

% 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')

```

Figure 2. MATLAB script to setup database connection. Comments are given in lines beginning with % and coloured green. Here users only need to specify the *namespace* and *Tenant* they wish to connect to which will be fixed for each project.

metadata in the archive annotation database. 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 3.4.

3.3. Data Archive (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 annotation tags 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, where the history of any changes to the data are recorded, as well as annotation. Both the data and metadata can be updated for new versions. A database with versioning enabled can enhance traceability when the data are tagged with the associated metadata [3]. The ability to trace software, file, and documentation changes through an unbroken chain (i.e., the version history) can help identifying bugs/errors and tracking system evolution. The ordered nature of versioning enables the user to return to a point of the evolution of the data and create a new branching point. This is particularly useful if errors have been identified or new methods have been developed, such as improved data processing.

The tool provides specific functions for: connecting to the database, uploading data following archiving as a HDF5 container file, 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 Figure 2.

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

```

% 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)

```

Figure 3. MATLAB script to upload local data to the database. Comments are given in lines beginning with % and coloured green. Users specify the location of the data they wish to upload (any number or format of data) in *local_dir* which also supports shortcuts, and the destination folder on the database in *database_dir*. The structure of folders on the database can resemble folder structures on computers which will not impact the searchability of the database.

(*database_dir*) in a function call. The function first creates an HDF5 container file that is populated with all the data in the directory specified by the user. Note this directory can contain any number of, or format of data files and also supports the use of shortcuts/links to other directories. The latter is vital when dealing with very large files that may be stored on multiple disparate devices, such as different laboratory instruments, and avoids the need to transfer data or duplicate data prior to using our tool. Next, the tool tags the container with the (well-formed XML) metadata as outlined in Section 3.2 allowing search functions in the database. Finally, this tagged HDF5 file is then uploaded to the database. The script for this is given in Figure 3 which highlights the simplicity of our tool's interface which is vital for non-expert users.

Similarly, data can be easily downloaded from the database. In this case "local_dir" is the folder location 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, i.e., there are no shortcuts when data are downloaded and unpacked. The script for downloading the data is given in Figure 4.

3.4. 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 Figure 5.

We can also search the contents 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,

```

% 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)

```

Figure 4. MATLAB script for downloading data from the database to the local computer. Comments are given in lines beginning with % and coloured green. Here *database_dir* is the data users wish to retrieve from the database and *local_dir* is the target directory on the local machine to download the data and automatically convert it back into its original format prior to upload to the database.

```

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

```

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

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), ensures that the data are accessible to all users in line with the GO FAIR principles [13]. 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.

4. RESULTS

We demonstrate our tool with a case study using a database from the Cancer Research UK Rosetta Grand Challenge (A24034) Project led by NPL's NiCE-MSI group [14]. 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 [3]. This database consists 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.

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 [15] making the data accessible.

Opening the database connection 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. We provide standard and simplified code for common queries allowing non-programmers to utilise this functionality. 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','SYNAPT G2-Si')`
- data from a particular sample (unique barcode from a separate sample management database, the data is integrated into 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. Example results for the conn.metadata_query('massRange', 'm/z 50-1200') query are given in Table 1.

As the data are contained in HDF5 containers with well-structured metadata all relevant associated datasets, including calibration files, processing scripts and the data are interoperable and re-usable, meaning external entities can more easily access, exchange and make use of the information contained within the HDF5 file.

5. CONCLUSIONS

We have introduced and demonstrated our tool that allows non-experts to interact with a well-curated database using the popular programming languages MATLAB and Python. By encapsulating multiple difficult APIs into a single, user-friendly application, the tool allows measurement scientists to easily ensure that their data is saved in a well-curated, FAIR-compliant, and traceable database without the need for specialised computational skills. The ability to collate all relevant data, tag the collated data with relevant machine actionable metadata, and upload it to a database with a single function call reduces the computational barrier significantly.

A single line of code is used to perform SQL-type searches on 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. One such benefit of the container file is the storage of data with associated metadata, processing and analysis codes providing interoperable and reusable data that is also traceable. We demonstrate this through a case study of data collected from a large-scale multi-site imaging project with large volumes of highly complex measurement data.

Template scripts further reduce this barrier and enable the capture of metadata 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.

ACKNOWLEDGEMENT

This work was funded by the Department for Business, Energy & Industrial Strategy through the UK's National Measurement System.

REFERENCES

[1] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton (+48 more authors), The FAIR Guiding Principles for scientific data management and stewardship, *Scientific Data* 6(1) 2019, 9 pp.
DOI: [10.1038/sdata.2016.18](https://doi.org/10.1038/sdata.2016.18)

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. Here we can see several objects from the same measurement site or of the same modality, or both, allowing inter-lab and multimodal studies and analysis. The file version also gives an indication of the provenance of the files and analysis.

ID	File Version	Measurement Site	Modality
Object A	1st	A	B
Object B	2nd	A	A
Object C	1st	C	A
Object D	4th	B	C
Object E	2nd	C	C
Object F	2nd	A	B
Object G	1st	C	C
...			

[2] S. A. Thomas, F. Brochu, A framework for traceable storage and curation of measurement data, *Measurement: Sensors*, 18(100201) 2021, 5 pp.
DOI: [10.1016/j.measen.2021.100201](https://doi.org/10.1016/j.measen.2021.100201)

[3] S. A. Thomas, F. Brochu, Curation at the point of measurement and traceability of measurement workflows, *Measurement: Sensors*, 23(1000399) 2022, 7 pp.
DOI: [10.1016/j.measen.2022.100399](https://doi.org/10.1016/j.measen.2022.100399)

[4] M. Santos, P. Sá-Couto, A. Silva, N. Rocha, In DICOM metadata-mining in PACS for computed radiography X-Ray exposure analysis: a mammography multisite study, *European Congress of Radiology-ECR 2014*, Vienna, Austria, 6-10 March 2014, 7 pp.
DOI: [10.1594/ecr2014/B-0276](https://doi.org/10.1594/ecr2014/B-0276)

[5] A. B. Haidich, Meta-analysis in medical research, *Hippokratia*, 14(Suppl 1): 29–37, 2010. Online [Accessed 17 March 2023]
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3049418/>

[6] N. Mikolajewicz, S. V. Komarova, Meta-Analytic Methodology for Basic Research: A Practical Guide, *Front. Physiol., Sec. Computational Physiology and Medicine*, 2019, 20 pp.
DOI: [10.3389/fphys.2019.00203](https://doi.org/10.3389/fphys.2019.00203)

[7] I. Bergeri, M. G. Whelan, H. Ware, L. Subissi, A. Nardone (+25 more authors), Global SARS-CoV-2 seroprevalence from January 2020 to April 2022: A systematic review and meta-analysis of standardized population-based studies, *PLOS Medicine*, 19(11): e1004107, 2022, 24 pp.
DOI: [10.1371/journal.pmed.1004107](https://doi.org/10.1371/journal.pmed.1004107)

[8] C. B. Joy, C. E. Adams, S. Lawrie, Haloperidol versus placebo for schizophrenia. *Cochrane Database of Systematic Reviews*, John Wiley & Sons, Ltd, 2001.
DOI: [10.1002/14651858.CD003082](https://doi.org/10.1002/14651858.CD003082)

[9] J. P. Higgins, I. R. White, A. M. Wood, Imputation methods for missing outcome data in meta-analysis of clinical trials. *Clinical Trials*. 2008;5(3), pp. 225-239.
DOI: [10.1177/1740774508091600](https://doi.org/10.1177/1740774508091600)

[10] N. Smith, D. Sinden, S. A. Thomas, M. Romanchikova, J. E. Talbott, M. Adegun, Building confidence in digital health through metrology, *The British Journal of Radiology*, 93(1109) 2020, 3 pp.
DOI: [10.1259/bjr.20190574](https://doi.org/10.1259/bjr.20190574)

[11] AWS, AWS S3 REST API protocol. Online [Accessed 17 March 2023]
<https://docs.aws.amazon.com/AmazonS3/latest/API/s3-api.pdf#Welcome>

[12] w3resource.com, Well-formed-XML. Online [Accessed 17 March 2023]
<https://www.w3resource.com/xml/well-formed.php>

[13] GO FAIR Int. Support and Coordination Office (GFISCO), GO FAIR Initiative. Online [Accessed 17 March 2023]
<https://www.go-fair.org/fair-principles/>

- [14] Cancer Research UK, Rosetta Project. Online [Accessed 17 March 2023]
<https://cancergrandchallenges.org/teams/rosetta>
- [15] A. Römpf, Th. Schramm, A. Hester, I. Klinkert, J.-P. Both, R. M. A. Heeren, M. Stoeckli, B. Spengler, imzML: Imaging mass

spectrometry markup language: A common data format for mass spectrometry imaging. *Methods Mol Biol.*, 696, 2011, pp. 205–224. DOI: [10.1007/978-1-60761-987-1_12](https://doi.org/10.1007/978-1-60761-987-1_12)