MOLECULAR IMAGING AND BIOLOGY

Mol Imaging Biol (2019) 21:1034–1043
DOI: 10.1007/s11307-019-01339-0

Published Online: 13 March 2019

WMIS World Molecular Imaging Society

Check for updates

RESEARCH ARTICLE

Data Curation for Preclinical and Clinical Multimodal Imaging Studies

Grace Gyamfuah Yamoah,[1] Liji Cao,[2] Chao Wu Wu,[3,4] Freek J. Beekman,[3,4] Bert Vandeghinste,[5] Julia G. Mannheim,[6] Stefanie Rosenhain,[1,7] Kevin Leonardic,[1,7] Fabian Kiessling,[1,8] Felix Gremse[1,7]

[1]*Institute for Experimental Molecular Imaging, Helmholtz Institute for Biomedical Engineering, RWTH Aachen University Clinic, Forckenbeckstraße 55, 52074, Aachen, Germany*
[2]*Inviscan SAS, Strasbourg, France*
[3]*MILabs B.V. Utrecht, Utrecht, The Netherlands*
[4]*Radiation Science & Technology, Delft University of Technology, Delft, the Netherlands*
[5]*Molecubes NV, Ghent, Belgium*
[6]*Department of Preclinical Imaging and Radiopharmacy, Werner Siemens Imaging Center, Eberhard Karls University Tuebingen, Tuebingen, Germany*
[7]*Gremse-IT GmbH, Aachen, Germany*
[8]*Comprehensive Diagnostic Center Aachen, RWTH Aachen University Clinic, Aachen, Germany*

Abstract

Purpose: In biomedical research, imaging modalities help discover pathological mechanisms to develop and evaluate novel diagnostic and theranostic approaches. However, while standards for data storage in the clinical medical imaging field exist, data curation standards for biomedical research are yet to be established. This work aimed at developing a free secure file format for multimodal imaging studies, supporting common *in vivo* imaging modalities up to five dimensions as a step towards establishing data curation standards for biomedical research.

Procedures: Images are compressed using lossless compression algorithm. Cryptographic hashes are computed on the compressed image slices. The hashes and compressions are computed in parallel, speeding up computations depending on the number of available cores. Then, the hashed images with digitally signed timestamps are cryptographically written to file. Fields in the structure, compressed slices, hashes, and timestamps are serialized for writing and reading from files. The C++ implementation is tested on multimodal data from six imaging sites, well-documented, and integrated into a preclinical image analysis software.

Results: The format has been tested with several imaging modalities including fluorescence molecular tomography/x-ray computed tomography (CT), positron emission tomography (PET)/CT, single-photon emission computed tomography/CT, and PET/magnetic resonance imaging. To assess performance, we measured the compression rate, ratio, and time spent in compression. Additionally, the time and rate of writing and reading on a network drive were measured. Our findings demonstrate that we achieve close to 50 % reduction in storage space for μCT data. The parallelization speeds up the hash computations by a factor of 4. We achieve a compression rate of 137 MB/s for file of size 354 MB.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s11307-019-01339-0) contains supplementary material, which is available to authorized users.

Correspondence to: Felix Gremse; *e-mail:* fgremse@ukaachen.de

Conclusions: The development of this file format is a step to abstract and curate common processes involved in preclinical and clinical multimodal imaging studies in a standardized way. This work also defines better interface between multimodal imaging modalities and analysis software.

Key Words: Data curation, Reproducibility, Credibility, Timestamp, Multimodal imaging, Metadata, Compression, Cryptographic hashing, File format, Serialization

Introduction

Medical imaging visualizes and records parts of the human or animal body, tissues, or organs of interest for purposes of clinical and preclinical research, diagnosis, and treatment among others [1, 2]. Each imaging modality (*e.g.*, x-ray computed tomography (CT), positron emission tomography (PET), single-photon emission computed tomography (SPECT), magnetic resonance imaging (MRI), and medical ultrasound (US)) has its own strengths and limitations. Multimodal imaging integrates the strengths of two or more modalities, while overcoming their individual limitations in comprehensively describing disease pathophysiology [3]. Using multimodal imaging techniques often allows for better image analysis and quantification. Figure 1 depicts the general multimodal imaging concept that consists of underlay (for anatomical details) and overlay (for functional or molecular details).

Data curation is the active management of data throughout its lifecycle of interest and usefulness to scholarly and educational activities ensuring data is available for discovery and reuse. For instance, in accordance with rules of good scientific practice and as a foundation for quality assured research data, the German Research Foundation (DFG) requests that primary data must be stored for ten years in sustainable and secure repositories at the institution where it was collected or in a nationwide infrastructure [4–6]. Cipra and Taubes in their separate publications in Science suggest the use of digital timestamps to prove that a given document existed at a particular time and also assure data integrity, all in an attempt to enhance credibility of published scientific data [7, 8].

Different imaging modalities store their acquired images in various data formats such as DICOM, NIfTI, *etc.* These formats may not be sufficient for efficient data manipulation, analysis, and image processing. Similarly, manufacturers of medical imaging systems use different proprietary formats to store images in digital form. The problem of different file formats is even bigger for preclinical imaging devices. These format differences pose significant challenges for

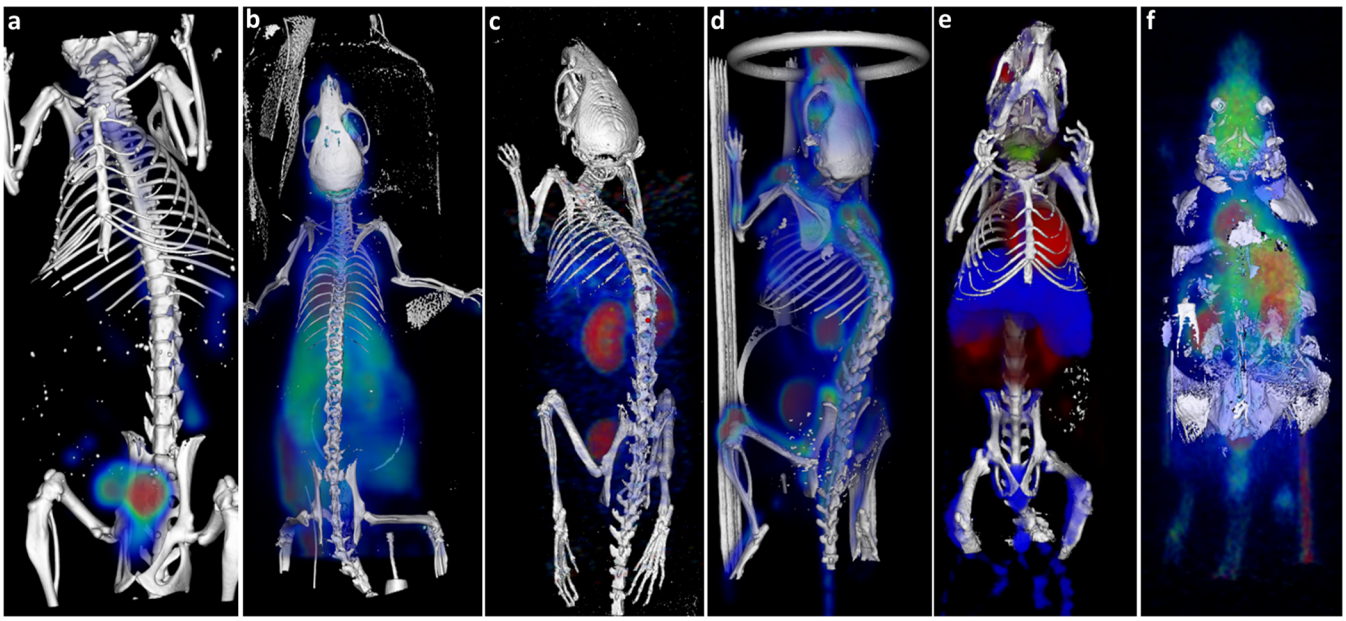

Fig. 1 Multimodal imaging datasets from several devices. **a** Fused CT (CT-Imaging, Erlangen, Germany) and fluorescence-mediated tomography (FMT) (PerkinElmer, Waltham, USA). **b** Integrated PET-CT (Inviscan, Strasbourg, France). **c** Fused SPECT and CT (Molecubes, Ghent, Belgium). **d** Integrated PET-CT (Trifoil, Chatsworth, USA). **e** Multispectral SPECT-PET-CT image (MILabs, Utrecht, the Netherlands) with three tracers. Blue: SPECT tracer 99Tc-MDP, red: PET tracer ^{18}F-FDG, green: SPECT tracer 123I-NaI. **f** Sequentially fused PET (Siemens Healthcare, Knoxville, TN, USA) and T2-weighted MRI sequence (Bruker, Ettlingen, Germany).

multimodal imaging studies particularly in registration, fusion, analysis, and curation of image datasets. Another difficulty with the difference in formats occurs if the utilized analysis software becomes obsolete, inaccessible, or undergoes repeated substantive changes that are not backward compatible [9, 10]. Post-processing software tools for kinetic modeling, spectral unmixing, and relaxometry require exact information about time points and channels. Our developed file format seeks to serve as the necessary bridge between imaging devices and multiple analysis software tools and to enhance post-processing.

Registration challenges mostly arise from the different physical principles that individual modalities are based on and the different complementary information they capture, mismatch in voxel data size and their dimensions, image translation, rotation, and incorrect mapping estimation between both images (*i.e.*, underlay and overlay) as well as from the different proprietary image formats used. We show few typical registration problems and some steps employed in fixing them in Fig. 2 using our data format. As a solution, a transformation must be adopted to ensure proper registration [11–13]. Metadata controls most steps of the curation process, from preservation to access and reuse [8, 10].

The aim of our study was to provide the research imaging community with a free secured file format for multimodal imaging that is compatible with all *in vivo* imaging modalities for five dimensions, supports multichannel and multitime series data, and equipped with the possibility to write cryptographically hashed images with or without trusted timestamp to file. This helps to improve the integrity of study data and promotes reproducibility and continuity. We developed this new image format in cooperation with users and providers of imaging hardware and software.

Materials and Methods

Content of the File

Many of the current imaging modalities are able to acquire multiple images successively over time, where the time is considered as the fourth dimension. Time points may be equidistant as in μCT scanners and ultrasound devices or non-equidistant as in PET scanners [14]. In addition, some of these devices are able to acquire multiple channel information, considering number of channels as the fifth dimension, as in the use of dual-energy CT [15], multichannel fluorescence molecular tomography (FMT) [16], MRI, multi-isotope SPECT, or use of different scanning protocols successively. It is desirable to have a single image file format that supports both isotropic (*e.g.*, usually provided by μCT and FMT scanners) and anisotropic (*e.g.*, in ultrasound and MRI modalities) voxels, multiple time points, and multiple channels. We propose a simple image file format with extension "gff" that writes and reads up to five-dimensional images to and from a single file. It consists of a data structure that serially stores voxel dimensions, sizes, and type, rotation and translation data elements, time and channel information, compression information, as well as metadata. We maintain both structured metadata (*e.g.*, voxel size, time point, and channels) and unstructured or more general metadata represented as key-value pairs (such as standardized uptake values (SUV), weight, age, *etc.*) in the format. The unstructured metadata can be easily deleted to ensure anonymity without negatively affecting the data needed for analysis. The format is capable of storing voxel data of any type ranging from char, half, float, double, signed, and unsigned short integer through to long integer,

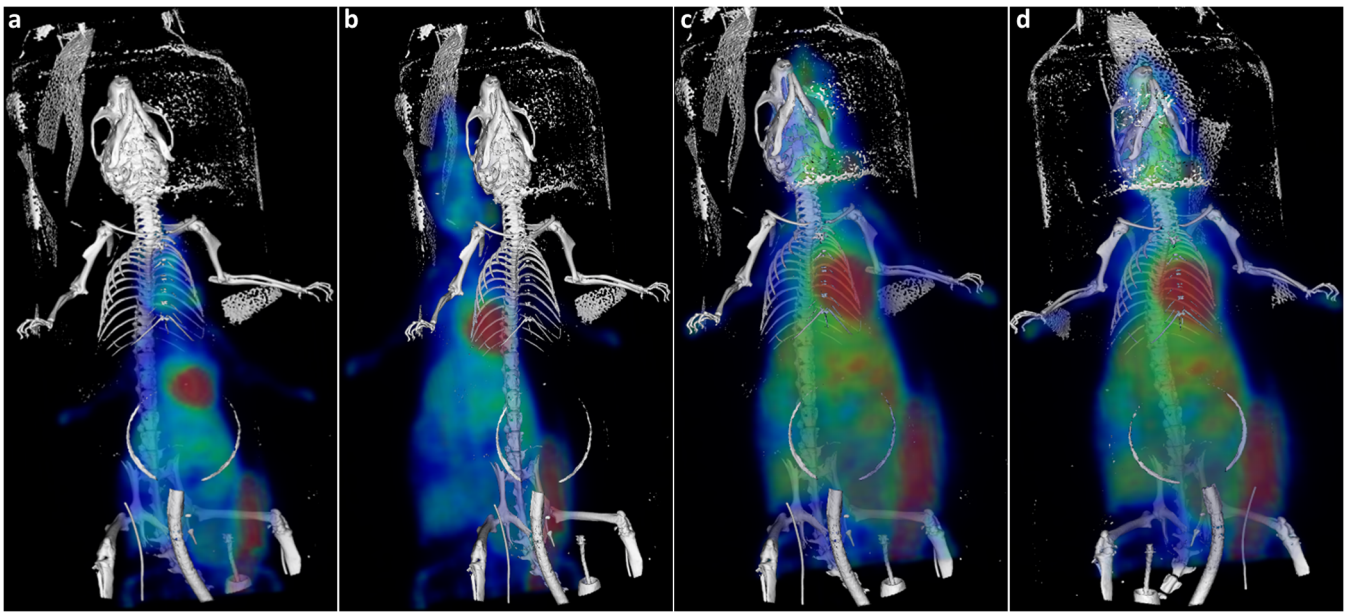

Fig. 2 Geometric transformation. **a** CT scan and PET scan wrongly positioned and mismatching voxel size. **b** Corrected voxel sizes. **c** Overlay centered on underlay. **d** Underlay flipped.

representing the data size per voxel. The X, Y, and Z dimensions provide the orthogonal spatial dimensions while dimensions T and C define time and channel dimensions respectively in this work. The *X*, *Y*, *Z*, *T*, and *C* dimensions are represented by 64-bit signed integers to support very large images. The voxel size is of type double and is measured in millimeters. As an option, we store scale factor and offset values to be applied to the voxel intensities. These values are stored as part of the structure. If these attributes are set to true, then the voxel intensities should be converted to float type by the software that reads the data. Our format stores all data in little endian format on the file.

Geometric Transformation

In order to analyze a given image dataset, a geometric transformation between voxel space and world space is needed. This transformation is used for co-registration of images obtained from two different modalities. The proposed file format supports the use of a rigid body transformation, a subset of the general affine transformations. In rigid body transformations, voxel sizes dictate the geometric scaling to be used, solving the scaling problems encountered in the use of rotation, scaling, and translation (RST) transformations. To move from data coordinate to world coordinate, we apply the formula below:

$$\begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = \boldsymbol{R} \begin{bmatrix} v_x * d_x \\ v_y * d_y \\ v_z * d_z \end{bmatrix} + \boldsymbol{t}$$

where R is an orthonormal 3×3 rotation matrix; d and v are voxel indices and sizes to be transformed by t, a 3×1 translation vector. If R is identity and t equals 0, then the world coordinate is determined by the product of voxel indices and sizes.

We provide two fields of type double in the file format that store the nine elements of the rotation matrix R and the three elements in the translation vector t. An inverse operation of the equation converts world coordinates to voxel coordinates. World position is provided in millimeters, while the voxel position is provided in a manner similar to numbering of the voxels with X being the fastest dimension and Z the slowest.

Time Point and Channel Information

The format stores the different time points at which the devices capture 3D volumes, a characteristic feature of four-dimensional images. Both equidistant time points and non-equidistant time points are supported. The metadata saved for each time point corresponds to the center time point and duration of each frame. We store the channel centers and widths as double data types. The dimension representing the number of time points and channels is stored as 64 bit integer. The field sizes used in storing the time point information and channel information depend on the number of time points and channels used. We support both equidistant and non-equidistant channel step information in the format.

Unstructured Metadata

We enable users to store unstructured metadata such as PET-CT or SPECT-CT SUVs like body weight, age, sex, *etc.* as key-value pairs. The metadata can be of any data type but is stored as key value pairs of type string. We define a data dictionary of type string, which contains entries that describe the unstructured metadata. The dictionary helps to check inside the operator[] key for validity. These metadata (strings) are encoded as Unicode in utf-8 encoding. Similarly, we support file and folder names in Unicode.

Compression

The proposed file format supports raw data (*i.e.*, the uncompressed voxel data) and zlib compression. We chose zlib due to its independence of CPU type, operating system, file system, and character set; its usability for interchange; and also due to its free usage without any patent issues [17]. In using the zlib compression method, we suggest using compression level 2 as a compromise between compression speed and compression factor. Each slice of the volume is compressed separately to enable parallel compression and decompression. Then, these compressed slices, their respective starting addresses and their individual lengths in memory, are serially saved to disk in a single file as shown in Fig. 3.

Cryptographic Hashing

Cryptographic hash functions apply a one-way algorithm on an arbitrary length of input data to produce a fixed length output. The hash function is a powerful tool, whose application helps protect the authenticity and integrity of information. We use the one-way SHA-256 algorithm to compute a cryptographic hash of the entire file. We compute all hashes in parallel, resulting in negligible overhead. We further implemented timestamp functionality based on the OpenSSL cryptographic library such that timestamps are appended to files written in our proposed "gff" file format. The cryptographic hashed data is sent as a timestamp request to our chosen TSA (*e.g.*, Bundesdruckerei, DFN timestamp server, *etc.*), which upon receipt, generates the timestamp and sends it back as a response as per the definitions of RFC 3161 [18]. The timestamp can be verified locally, but its creation may not be possible locally since it requires internet connectivity. Due to this internet connectivity requirement, the inclusion of timestamp is optional.

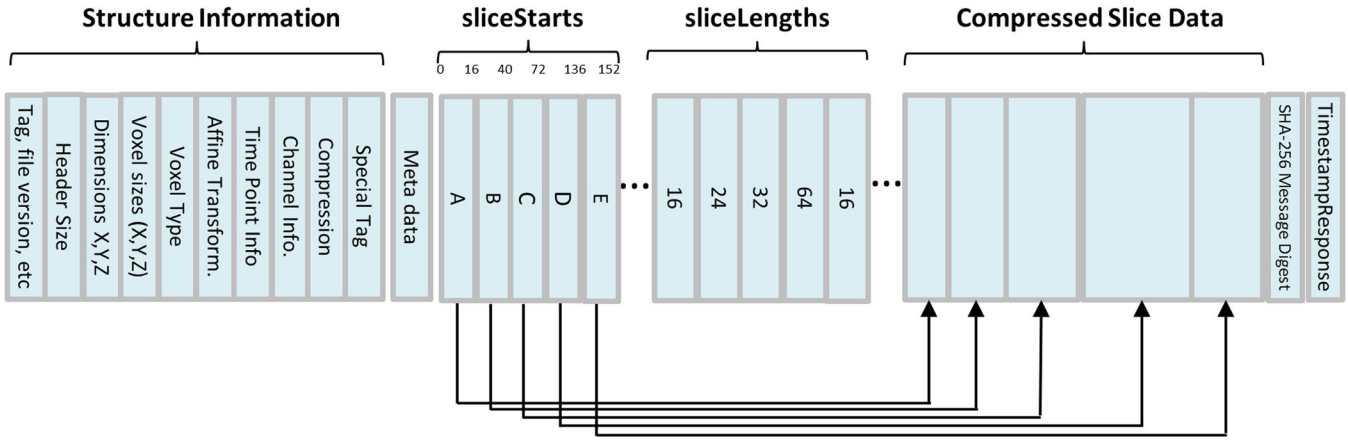

Fig. 3 Memory layout of the gff file format. Data pertaining to the fields in the structure, compressed data slices, their hashes, and the optional timestamp are all serially written to the file.

Memory Layout of File Format

The fields or objects in the file structure include versioning numbers, header size, dimensions and sizes of voxels, voxel data type, affine transform elements, time dimensions, channel dimensions, compression information, among others, and these are represented in the header structure as shown in Fig. 3. A 32-bit tag is placed at the beginning of the structure (0xD8CA0B00) as a signature that uniquely identifies the file type. The same is placed at the end of the structure, but only as a control. It is important to note that the memory space allotted to the header for each structure depends on the number of dimensions of the structure. Data pertaining to all these fields are written to the file serially. In addition, a metainformation field that stores a map of key-value pairs is serialized to the structure. The address that points to the beginning of each slice (sliceStarts) and also the length of each slice (sliceLengths) are saved in their respective vectors and are serially saved in memory. The vectors allow for reading a subset of slices. The sliceStarts vector with addresses correspond to the actual compressed slices of the image data acquired that are serially written to the file. The sliceLength vector provides the size in bytes of each slice. These two vectors (sliceStarts and sliceLength) have a certain degree of redundancy on purpose to allow serialization of slices in arbitrary order, *e.g.*, the order of generation which also simplifies parallel encoding and decoding of individual slices. Finally, the hash and optional timestamps are appended to the end of the file. Figure 3 gives a pictorial representation of the memory layout of the proposed file format.

C++ Implementation

The file format's template-based implementation is in two simple C++ files with a little over 1000 lines of code which are available as supplemental material for unrestricted use following a MIT-style license. We provide detailed explanation on the implementation and resources used in the Supplementary Material (Online resource 1). Figure 4 presents a code snippet showing usage of the implementation.

Example Images Used for the Experimentation and Evaluation

Two image datasets from five different mice were used. These are 3D μCT image datasets and segmentation image datasets as shown in Fig. 5. The images were obtained from experiments conducted for assessing the sensitivity and accuracy of hybrid fluorescence-mediated tomography in deep tissue regions [19]. The average sizes of the μCT and segmentation image datasets are 338 MB and 169 MB, respectively. The μCT scans and segmentation files were visualized and analyzed using the Imalytics Preclinical software (Gremse-IT GmbH, Germany) [20].

Results

Performance Measurements

We examined the time spent on compressing and decompressing the experimental files, and also time required for writing and reading on the network drive. The test was carried out for the ten compression levels specified in zlib ranging from 0 to 9. As shown in Fig. 6a, at compression level 0, no compression takes place while the time taken to complete compression at the other levels increases with increasing compression levels in both experimental file types. In Fig. 6b, we observe that the compression (referring to Fig. 6a) resulted in the decrease in total saving time as compared to level 0, because more time is saved in disk IO compared to the additional time spent for compression. Based on these results, level 2 was chosen as default since it appeared to be a good compromise for both speed and size. The decrease in saving time can be a reflection of the reduced file sizes at the different compression levels as shown in Fig. 6a.

```
//Create the structure
Structure5D s;
s.voxelType = VoxelTypeFLOAT;
s.dimX = 101;
s.dimY = 102;
s.dimZ = 31;
s.voxelSizeX = 0.5;
s.voxelSizeY = 0.6;
s.voxelSizeZ = 0.7;

GFF::Meta meta("System");
meta["Weight"] = "83.0";
meta["Age"] = "66";
meta["Scanner"] = "Scanner12";
meta["Software Version"] = "1.0.2";
meta["Camera Pixel Size"] = "(um)=75.0";
meta["Camera X/Y Ratio"] = "1.0002";
//load meta data

GFF::Meta meta1("Acquisition");
meta1["Filename Prefix"] = "Maus_";
meta1["Number Of Files"] = "814";
meta1["Camera binning"] = "1x1";
meta1["Image Rotation"] = "-0.03400";
meta1["Image Format"] = "GFF";
//load meta data

GFF::Meta meta2("Reconstruction");
meta2["Program Version="] = "Version: 1.6.10.2";
meta2["Reconstruction from ROI"] = "ON";
meta2["Filter type description"] = "Hamming (Alpha=0.54)";
//load meta data

std::vector<GFF::Meta> metas;
metas.push_back(meta);
metas.push_back(meta1);
metas.push_back(meta2);

StepInfo timeInfo(dimT);
//load time dimension data
s.timeInfo = timeInfo;

StepInfo channelInfo(dimC);
//load channel dimension data
s.channelInfo = channelInfo;

float* data = new float[s.dimX * s.dimY * s.dimZ * s.DimT() * s.DimC()];
//fill in voxels

bool createTimestamp = false;

//save the image
GFFIO().SaveImage5D(data, s, metas, fileName, createTimeStamp);
```

Fig. 4 Usage example of new format. A C++ class "Structure5D" is defined to represent the five-dimensional image structure with the required fields (*X*, *Y*, and *Z*) including time and channel dimensions (dimT and dimC). We instantiate an object of this class, *s*, populate its fields, provide metainformation, and write the structure and compressed slices to file.

We also performed a comparative study to identify the costs involved in saving with and without cryptographic hashes. From the graph in Fig. 6d, we observed that saving with hash is computationally more expensive compared to saving without hashes. However, from our experiments, we found out that the margin of difference between saving with the cryptographic hash and saving without it was almost negligible, not significant ($P > 0.05$). Hence, considering the integrity benefits that cryptographic hashes provide, we suggest saving with hash to achieve the set security goal.

Image Compression Measurements

The experimental image files were compressed using zlib compression algorithm at level 2, compensating for both

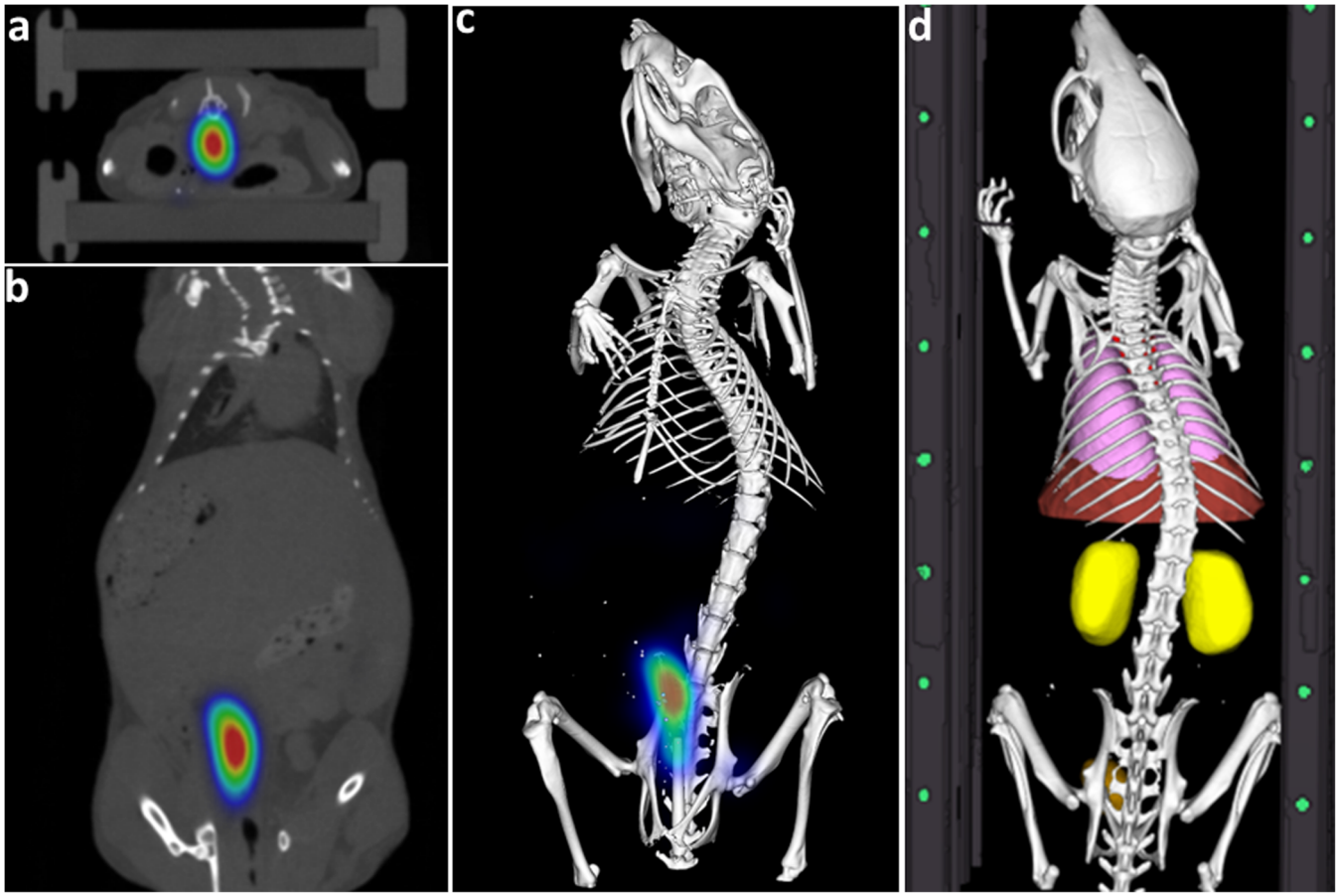

Fig. 5 Image datasets used for experimentation and evaluation. µCT dataset with FMT reconstruction overlay showing the mouse in (**a**) transversal and (**b**) coronal views. **c** 3D view of the mouse with rectal insertion and (**d**) the mouse in a mouse bed with various segmented organs surrounded by markers.

speed and size. zlib library permits the use of several compression options. We explored the DEFLATE and HUFFMAN-only options shown in Table 1. The table shows the sizes of the experimental input files prior to compression and their resulting sizes after compression, aiding in the computation of the compression ratios. From the Table 1, we noticed that the µCT image file is almost 50 % compressed while the segmentation file is over 99.6 % compressed. Interestingly, we observed that the HUFFMAN-only option yields 4× better compression rate for µCT files but fails to achieve compression ratio as good as that which was achieved by the Deflate option. The decision to choose which of these options rests with the user depending on the compression goals set, speed, or size.

Discussion

In this work, we proposed and developed a simple image file format that is usable for all *in vivo* imaging modalities generating volumetric datasets with regular grids. This new file format is particularly suitable for multimodal imaging including CT-FMT, PET-CT, SPECT-CT, and PET-MRI. We provided an implementation with few dependencies from the zlib library, OpenSSL library (optional), and Oliver Gay's version of sha256 [17, 21]. The software is platform-independent. The simple implementation is contained in two C++ header files. The file format was developed in close collaboration with device manufacturers including Inviscan (Strasbourg, France), Molecubes (Ghent, Belgium) and MILabs (Utrecht, the Netherlands) and a software supplier Gremse-IT (Aachen, Germany) to improve the interface between imaging hardware and analysis software.

The generation and/or collection of multidimensional medical image sequences such as *in vivo* µCT images, functional magnetic resonance imaging (fMRI), *etc.* is increasing at an alarming rate and necessitates structured and secured measures for managing, accessing, preserving, and reusing this huge amount of data since such digital data are typically not stored in long-term institutions such as libraries [6, 22, 23]. It also requires considerable amount of memory storage space in order of several gigabytes per acquisition, posing storage difficulties. However, image data has different types of redundancies which compression algorithms (both lossy and lossless) take advantage of, in

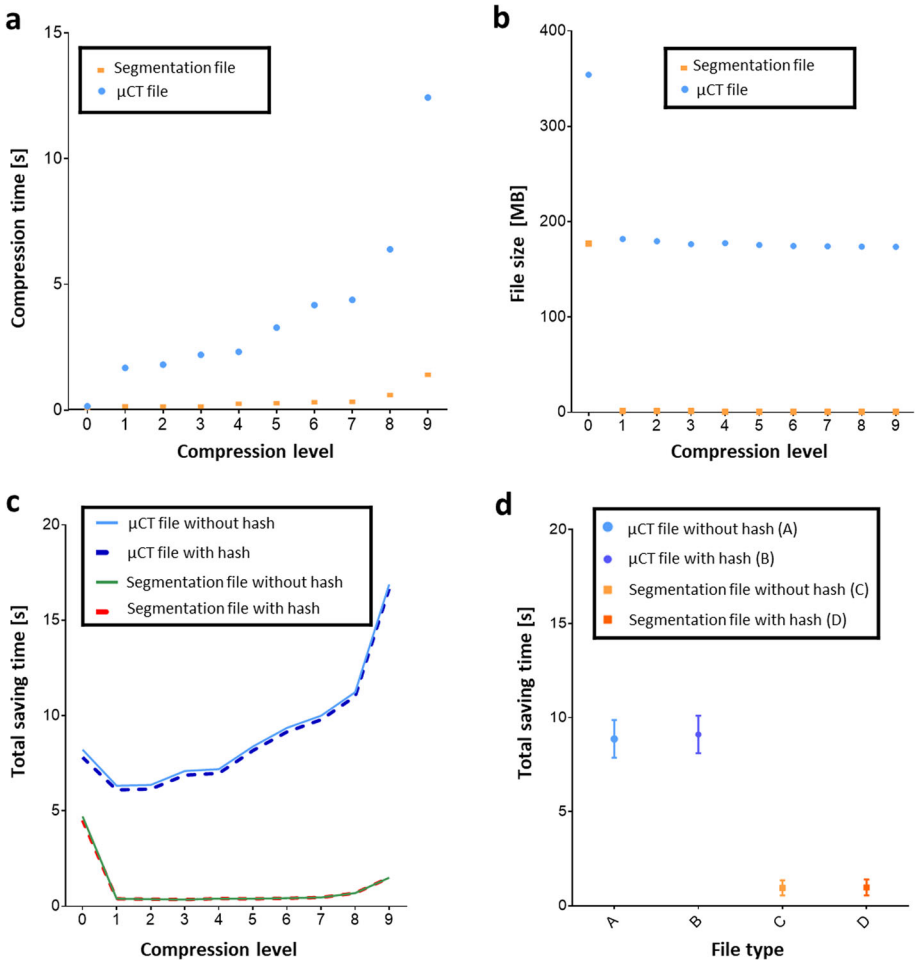

Fig. 6 Compression time measurements, compressed sizes, and an analysis on saving with and without hash. **a** Time spent in compression for both µCT and segmentation files as function of the zlib compression level. Level 0 records the least compression time for both files, (**b**) file sizes obtained for all compression levels, and (**c**) amount of time spent in saving both µCT and segmentation files with and without cryptographic hashes to our network drive. Compression levels 1 and 2 are good compromises and result in lowest disk saving times and (**d**) comparison of saving time of non-hashed files to hashed files. The saving times for hashed and non-hashed files were compared using repeated measures one-way ANOVA with Tukey post-test (*P* value of 0.05). The saving times of **b** and **d** increased slightly but not significantly with $P > 0.05$ in both cases.

order to increase the effective data densities on storage devices and optimize transmission costs [24–26]. Lossy compression algorithms, though results in smaller sizes, fail to be the choice for clinical use and preclinical research since they may eliminate certain critical information needed for diagnosis, analysis, and legal purposes [27, 28]. In this work, we employ the lossless compression implemented in the widely adopted and freely available zlib library. File formats characterized by properties such as complete and open documentation, platform-independence, and lossless compression among others have been identified to have the capability of preserving their contents and functionality over a long-term [29, 30].

Metadata controls most steps of the curation process, from preservation to access and reuse [5, 31]. However, its usage raises concerns about anonymity in clinical research

Table 1. File compression analysis using DEFLATE option and HUFFMAN-only option. Sizes of experimental input files prior to and after compression at level 5 and corresponding ratios and rates of compression

Image file type	Input file size (MB)	Compression option	Output file size (MB)	Compression ratio	Compression rate (MB/s)	Decompression rate (MB/s)
µCT	354.0	Deflate	175.49	2.01	105.0	921.3
		Huffman-only	201.4	1.760	491.2	1141
Segmentation	177.0	Deflate	0.63	280.9	745.3	1569
		Huffman-only	22.4	7.890	598.6	1810

since it may include subject's personal information and other traceable information that could raise privacy concerns [32]. Due to this and other complexities, certain image file formats such as NIfTI-1 provide limited support for the storage of the various image acquisition parameters as opposed to what is available in DICOM [33]. Also, writing a single large file to disk is comparatively more efficient and cost-effective than writing several small files to disk. [34]. Hence, in our file format, data is written as single file per modality as opposed to other file formats.

To assess the performance of our file format, we measured the rates, time, and ratios of compression and decompression and also measured time and rates of both writing and reading on a network drive. We took particular interest in the network drive analysis since it happens to be the most heavily used drive in research centers, institutes, and clinical centers. We recorded an average writing and compression rates of 28 MB/s and 137 MB/s, respectively on our network drive, showing that writing an uncompressed file to a network drive is more time consuming than compressing the same file. Also, the disk reading and decompression rates recorded were 97 MB/s and 1288 MB/s, respectively, indicating that the decompression is approximately 13 times faster than the disk reading speed and the compression is approximately 4 times faster than the writing speed. Both the hashes and compression/decompression rates are computed in parallel to minimize duration of writing files and enhance efficiency. We achieve a hashing speed of 822 MB/s. The parallel compression achieves a speedup by factor of 2 and 3 for the μCT and segmentation files, respectively.

It is worth mentioning that the reading and writing rates are affected if multithreaded parallel processing is in use or not. To further enhance efficiency, GPUs can be used for the computation.

The proposed file format can store 3D data, multiple time series, and multichannel data resulting in what we refer to as five-dimensional data. Future extensions of the format may provide more dimensions to cater for time intervals and dimensions in ECG-gated images, respiratory-gated images, and dual-gated images, among others.

Incremental reading and writing of slices are enabled with the format. In addition, the files in this proposed format are easy to anonymize by changing the file name and removing key-value pairs constituting the unstructured metadata without the danger of losing relevant information. This provides privacy protection for study participants, concealing sensitive information that could easily be traceable. We provide a dictionary-based mechanism to store predefined keys for the key-value pairs to enhance standardization.

A unique feature of our format is the inclusion of trusted timestamps. Timestamps provide a legally accepted way to prove the existence of a file at a certain time. Timestamps could aid in ensuring credibility and reproducibility, properties that are crucial to scientific research [7, 8, 35]. The RFC 3161 specification states that only a hashed representation of the data or file should be time-stamped to avoid unnecessary original data exposure [18]. Timestamps are typically issued by trusted third parties or timestamp authorities (TSA). There are free timestamp servers for academic use [36] and commercial servers whose service comes with a cost. For instance, the D-STAMP timestamp, provided by the German Bundesdruckerei (Federal Bureau for Printing), costs approximately 10 cents. The creation and checking of timestamp introduces some overhead in the writing and reading of files, respectively. However, considering the benefit of the timestamp and even the negligible time overhead for hash creation observed in our measurements, it makes sense to go for the timestamp and achieve the integrity goal. It is important to note that the inclusion of the timestamp is optional and could be added later (*e.g.*, in cases where internet connectivity is a problem). We also make the source code and sample datasets freely available for use and reproducibility purposes and also as a step in cutting down on replication costs [37]. Reproducibility of science is an aspect of credibility that requires researchers to provide proper and sufficient information as well as accurate documentation to enable the verification of their work [38, 39]. In recent times, it has been observed by some researchers [40], the National Institutes of Health (NIH) [38], and other government funding bodies that there is a growing irreproducibility of science and poor data management. This may be incidental or deliberate [41]. Data curation methods and tools could help restore confidence and trust in scientific research and subsequently enhance credibility [31]. The format provides for backward compatibility support in the event of versioning change.

Conclusion

In conclusion, a free file format and a software tool for multimodal imaging studies that seek to make post-processing easier, solving the vendor variability, and data import issues, particularly for complex image analysis methods have been developed. The development of this file format is a step to archive and curate preclinical and clinical scientific imaging studies in a standardized way. Find the summarized features of the file format in Suppl. Table 1.

Funding information. This research was supported by the German Academic Exchange Service (DAAD), German Research Foundation (DFG; GR 5027/2-1), German Higher Education Ministry (BMBF) (Biophotonics/13 N13355), Federal Government of North-Rhine Westphalia (EFRE), and the European Union (FP7), and a grant from the Interdisciplinary Centre for Clinical Research within the faculty of Medicine at the RWTH Aachen University (E8-13).

Compliance with Ethical Standards

Conflict of Interests

F. Gremse is the owner of Gremse-IT GmbH. F. Beekman holds shares in MILabs B.V. B. Vandeghinste holds shares in Molecubes NV.

References

1. Bourne R (2010) Fundamentals of digital imaging in medicine. Springer Science & Business Media, London
2. Hill DLG, Batchelor PG, Holden M, Hawkes DJ (2001) Medical image registration. Phys Med Biol 46:R1–R45
3. Lee SY, Jeon SI, Jung S, Chung IJ, Ahn CH (2014) Targeted multimodal imaging modalities. Adv Drug Deliv Rev 76:60–78
4. DFG, Ger Res Foundation—Handling of research data. http://www.dfg.de/en/research_funding/proposal_review_decision/applicants/research_data/index.html. Accessed 25 Feb 2019
5. Ray JM (2014) Introduction to research data management. In: Ray JM (ed) Research Data Management: Practical Strategies for Information Professionals. Purdue University Press, West Lafayette, pp 1–22
6. Lord P, Macdonald A, Lyon L, Giaretta D (2004) From Data Deluge to Data Curation. Conference: Proceedings of the UK e-Science All Hands Meeting 2004, pp. 371–37
7. Cipra B (1993) Electronic time-stamping: the notary public goes digital. Science 261:162–163
8. Taubes G (1994) Technology for turning seeing into believing. Science 263:318–318
9. Hedstrom M (1997) Digital preservation: a time bomb for digital libraries. Comput Hum 31:189–202
10. Abrams S (2007) DCC digital curation manual Installment on file formats. HATII, University of Glasgow; University of Edinburgh; UKOLN, University of Bath; Council for the Central Laboratory of the Research Councils. https://www.era.lib.ed.ac.uk/handle/1842/3351. Accessed 27 Feb 2019
11. Maes F, Vandermeulen D, Suetens P (2003) Medical image registration using mutual information. Proc IEEE 91:1699–1722
12. Brown LG (1992) A survey of image registration techniques. ACM Comput Surv 24:325–376
13. Zitová B, Flusser J (2003) Image registration methods: a survey. Image Vis Comput 21:977–1000
14. Pöschinger T, Renner A, Eisa F, Dobosz M, Strobel S, Weber TG, Brauweiler R, Kalender WA, Scheuer W (2014) Dynamic contrast-enhanced micro-computed tomography correlates with 3-dimensional fluorescence ultramicroscopy in antiangiogenic therapy of breast cancer xenografts. Investig Radiol 49:445–456
15. Gremse F, Grouls C, Palmowski M, Lammers T, de Vries A, Grüll H, Das M, Mühlenbruch G, Akhtar S, Schober A, Kiessling F (2011) Virtual elastic sphere processing enables reproducible quantification of vessel stenosis at CT and MR angiography. Radiology 260:709–717
16. Kunjachan S, Gremse F, Theek B, Koczera P, Pola R, Pechar M, Etrych T, Ulbrich K, Storm G, Kiessling F, Lammers T (2013) Noninvasive optical imaging of nanomedicine biodistribution. ACS Nano 7:252–262
17. Gailly J, Adler M (2004) zlib compression library. http://www.dspace.cam.ac.uk/handle/1810/3486. Accessed 27 Feb 2019
18. Adams C, Cain P, Pinkas D, Zuccherato R (2001) Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP). RFC 3161. https://doi.org/10.17487/RFC3161
19. Rosenhain S, Al Rawashdeh W, Kiessling F, Gremse F (2016) Sensitivity and accuracy of hybrid fluorescence-mediated tomography in deep tissue regions. J Biophotonics 10(9):1208–1216. https://doi.org/10.1002/jbio.201600232
20. Gremse F, Stärk M, Ehling J, Menzel JR, Lammers T, Kiessling F (2016) Imalytics preclinical: interactive analysis of biomedical volume data. Theranostics 6:328–341
21. Gay O (2007) Fast software implementation in C of the FIPS 180-2 hash algorithms SHA-224, SHA-256, SHA-384 and SHA-512. http://www.ouah.org/ogay/sha2/. Accessed 25 Feb 2019
22. Karasti H, Baker KS, Halkola E (2006) Enriching the notion of data curation in E-science: data managing and information infrastructuring in the long term ecological research (LTER) network. Comput Support Coop Work CSCW 15:321–358
23. Heidorn PB (2011) The emerging role of libraries in data curation and E-science. J Libr Adm 51:662–672
24. Gonzalez RC, Woods RE, Masters BR (2009) Digital image processing, third edition. J Biomed Opt 14:029901
25. Wong S, Zaremba L, Gooden D, Huang HK (1995) Radiologic image compression-a review. Proc IEEE 83:194–219
26. Carpentieri B, Pizzolante R (2014) Lossless compression of multidimensional medical images for augmented reality applications. In: De Paolis L, Mongelli A (eds) Augmented and Virtual Reality. Lecture Notes in Computer Science, vol 8853. Springer, Cham
27. Zukoski MJ, Boult T, Iyriboz T (2006) A novel approach to medical image compression. Int J Bioinforma Res Appl 2:89–103
28. Ström J, Cosman PC (1997) Medical image compression with lossless regions of interest. Signal Process 59:155–171
29. Mortimore J Guides: Data Management Services: Recommended file formats for long-term data curation. http://georgiasouthern.libguides.com/c.php?g=410908&p=2955521. Accessed 25 Feb 2019
30. File formats and standards - Digital Preservation Handbook. http://www.dpconline.org/handbook/technical-solutions-and-tools/file-formats-and-standards. Accessed 25 Feb 2019
31. Bird CL, Willoughby C, Coles SJ, Frey JG (2013) Data curation issues in the chemical sciences. Inf Stand Q 25:4
32. Doel T, Shakir DI, Pratt R, Aertsen M, Moggridge J, Bellon E, David AL, Deprest J, Vercauteren T, Ourselin S (2017) GIFT-cloud: a data sharing and collaboration platform for medical imaging research. Comput Methods Prog Biomed 139:181–190
33. dfwg NIfTI: — Neuroimaging Informatics Technology Initiative. https://nifti.nimh.nih.gov/. Accessed 25 Feb 2019
34. Thain D, Moretti C (2007) Efficient access to many small files in a filesystem for gridcomputing. In Proceedings of the 8th IEEE/ACM International Conference on Grid Computing (GRID '07). IEEE Computer Society, Washington, DC, pp 243–250. https://doi.org/10.1109/GRID.2007.4354139
35. Anderson C (1994) Easy-to-alter digital images raise fears of tampering. Science 263:317–318
36. Manouchehri D (2016) List of free rfc3161 servers. In: Gist https://gist.github.com/Manouchehri/fd754e402d98430243455713efada710. Accessed 25 Feb 2019
37. Gertler P, Galiani S, Romero M (2018) How to make replication the norm. Nature 554:417–419
38. Marcus E (2014) Credibility and reproducibility. Cancer Cell 26:771–772
39. Begley CG, Ioannidis JPA (2015) Reproducibility in science: improving the standard for basic and preclinical research. Circ Res 116:116–126
40. Baker M (2016) 1,500 scientists lift the lid on reproducibility. Nat News 533:452–454
41. Munafò MR, Nosek BA, Bishop DVM, Button KS, Chambers CD, Percie du Sert N, Simonsohn U, Wagenmakers EJ, Ware JJ, Ioannidis JPA (2017) A manifesto for reproducible science. Nat Hum Behav 1:s41562–016–0021–016