PREFACE

In the past four years, several independent workshops organized by the European Bioinformatics Institute (EBI) and the Research Collaboratory for Structural Bioinformatics (RCSB) at Rutgers University, have reached a community consensus that it is time to establish publicly supported, one-stop deposition and retrieval facilities for cryoEM density maps, atomic models and associated metadata.

In an idealized repository, cryoEM experts and biological end-users would build from what the previous researchers have contributed, forming the basis for subsequent hypothesis-driven research and knowledge discovery. Derivative data from such discoveries could also be deposited into the repository.

What is initially envisioned is an integrated file of 2D or 3D spatial image datasets along with relevant acquisition and reconstruction metadata; but other optional metadata, such as animations that temporally illustrate the spatial datasets or ISO standardized xml-based MPEG7 that concisely annotates images, could also enhance the understanding of the spatial image datasets and together dramatically improve their access. For example, with an integrated archive-ready 200 GB data file that contains 1) a large 3D cryo-electron tomogram of a biological structure, 2) with sub-sampled, hierarchically organized, compressed, chunked 3D tomograms, and also including 3) various animations and 4) MPEG7 meta-datasets; users could efficiently examine the animations through a simple web browser to get a first look before directly interacting with the 3D spatial image datasets. By defining a region of interest in the animation timeline, its camera positions could be extracted from the MPEG7, allowing one to create a region of interest projection needed to extract a small subset from the 3D images, beginning with the low resolution tomograms and progressing to high resolution. The MPEG7 could also be used to annotate 3D structural features of the tomogram as well as specific events and 3D structures in the animations.

The amalgamation of these datasets currently does not exist within a single data format. Significantly, there is no EM format that is optimal for managing very large images and is extensible enough to assimilate other communities' heterogeneous metadata. Unfortunately, no generic standardized definition of scientific imagery exists, although the astronomical community has come closest in regards to their information frameworks, disparate image modalities and complex data analysis.

In a complex research pipeline this lack of a robust and uniform scientific image has resulted in fragmented definitions, incompatible software, and redundant data files, the aftermath being confusing conversions and basic incompatibilities. Software tools for density map creation, map segmentation, hypothetical model assessment, visualization, and data integration have all been affected by this lack of simple image interchange and the near total disconnect of contextual metadata as it progresses through the pipeline. Without a comprehensive and uniform definition of a scientific image, efficient use of the archived data and derivatives will be an ongoing fumble. An urgent prediction is that if the images cannot be effectively designed and utilized for both archival and operational settings, the associated metadata has significantly less value.

Moreover, this comprehensive cryoEM data file must be designed in a manner that will be simple to describe, straightforward to maintain, and must be able to evolve as the scientific field grows and matures.

Finally, it is essential to seek out broad views, from both within and without the cryoEM and biological user communities, throughout the design and implementation so that the final system will incorporate the highest standards to serve the needs of both current and future scientists.

ImageCore v1-RFC1

INTRODUCTION

At this time, different biological research communities are beginning to examine and adopt the Hierarchical Data Format (HDF) version five, particularly for activities involving data acquisition, visualization, and image deposition. HDF is a high performance data storage and manipulation tool developed by the National Center for Supercomputer Applications to create and interact with large heterogeneous datasets. In addition to being open source, HDF offers a number of significant features that will be difficult if not impossible to replicate by any other method; therefore HDF5 is the obvious choice for managing large scientific images.

Because each research community has different priorities, these communities' metadata and related ontologies are frequently incompatible or irrelevant. But, there is one type of data that forms the coin of the realm: the scientific image. This basic form of information must be broadly and effectively defined in order to obtain compatibility across biological fields. A primary concern is that different biological communities will adopt HDF and will define image implementations that are inherently conflicted and mutually incompatible; all of which must be avoided in order to allow maximum scientific benefit from the data.

To attain a desirable outcome, first it is absolutely necessary that different biological communities exercise their ability within HDF to reserve their data domains, thereby avoiding namespace conflict; second it is crucial to establish best practices that will lead to uniform definitions of scientific images within HDF. By accomplishing these two tasks, the vast majority of scientific imaging by different biological communities will become co-existent and interpretable across inter-disciplinary research communities that use HDF.

Because of the limited use of HDF in biomedical research, the broad interest and need for data compatibility, and the simple nature of scientific images as implemented in HDF; the possibility of avoiding conflict and promoting cooperation is high.

DEFINITIONS

1) A scientific image is an n-dimensional array of homogeneous pixels.

2) A picture element, the pixel, frequently consists of a single scalar component, but may consist of multiple components or channels, whose scalar or non-scalar values are direct image measurements, or computational results derived from measurements or simulations.

3) Images and the pixels have core metadata that define organization, sizes, units, and coordinate systems.

At this point, the scientific communities begin to functionally, operationally and theoretically diverge as to their non-image metadata. It is necessary to explicitly link and assemble the user metadata and image metadata as an amalgamation of digital data by some means, the typical scenario being a single image file with an un-extensible header preceding the contiguous block of pixels, the alternate second largest approach is to have two files, one containing metadata and the other containing pixels. As the original image is circulated to other scientific communities, new and unique meta-datasets will be created and must be linked to maintain appropriate context and knowledge improvement, which is an Achilles heal of, if not all, scientific image data formats.

Please send comments to
Matthew Dougherty, matthewd@bcm.edu

ImageCore v1-RFC1

SPECIFICATION

ImageCore (IC) is a proposed methodology to organize images and arbitrary metadata within an HDF file. It can be used as a generic image format, or it can be linked into existing image data formats that implement HDF.

The key methods:

1) Define a reserved HDF namespace for image-pixels using a simple naming convention.
2) Associate images, pixels and metadata using the Resource Description Framework (RDF).
3) Provide an optional reserved IC namespace for grouping adjunct user or application metadata.
4) Provide integration protocols for images and metadata that exists in user or application namespaces.

There are four HDF imageCore elements:

1) An HDF root group "/imageCore/", forming the primary IC namespace.
2) A *chronological* RDF image log, having the reserved HDF dataset name "/imageCore/0". It is an nx4 extensible table of time-stamped RDF values:
   i) The first column contains ***time-stamps*** that log the RDF entry. The time-stamps conform to the standard *ISO 8601,* UTC date and time specification "*YYYY-MM-DDThh:mm:ss.sZ"*.
   ii) The second column contains RDF image ***subjects***. The subject is a positive integer corresponding to an image dataset name that is located in the IC namespace "/imageCore/".
   iii) The third column contains RDF image ***predicates.*** There are three basic categories: IMAGE, LINK, and LOG.
   iv) The fourth column contains RDF image ***objects***.

3) An expandable number of N-dimensional images whose pixels are contained in HDF datasets. These pixel datasets have reserved HDF dataset names corresponding to positive integers that have no preceding zeros or symbols, for example "/imageCore/1", "/imageCore/2", and "/imageCore/314159". Image datasets may also exist in non-IC namespaces and these HDF pixel datasets will have naming conventions defined by application programs or user selections; in this case, the pixel datasets are linked into the "/imageCore/" group so as to have names corresponding to positive integers, functionally producing the same effect of having the images created in the IC namespace. The positive integer image dataset names were selected to enforce bare simplicity and to maintain a neutral naming convention that is more easily interpretable across various languages and cultures.

4) Optional metadata can be grouped in the IC namespace corresponding to the image datasets names. These adjunct metadata may contain any type of user datasets, groups, and attributes, such as CCP4 image format headers, MPEG7 annotation datasets, segmentation masks, or community specific metadata such as cryo-EM acquisition and reconstruction parameters. The names of these adjunct metadata are a variant of the associate image dataset name by appending an asterisk to the name to create an adjunct group, for example "/imageCore/1*/animation.mpg", "/imageCore/4*/camera.mp7", "/imageCore/2*/OME.xml", "/imageCore/3*/1atn.pdb", or "/imageCore/4*/EMAN2/zz98.xml". Metadata that are in non-IC namespaces can be associated to IC images through "/imageCore/0" RDF.

There are other minor HDF attributes that are attached to the HDF objects to make them compliant with existing HDF image and python strategies. These are referenced in the HDF5 Image and Palette Specification version 1.2, and appendix E of the Pytables version 2.0 user manual.

Please send comments to
Matthew Dougherty, matthewd@bcm.edu

APPENDIX: TABLES

/imageCore/0 RDF description

TIMESTAMP date&time	SUBJECT + integer	PREDICATE VL string	OBJECT VL string	multi defined	optional	OBJECT Description
ISO 8601	image ID	IMAGE.chunking	integer 1D array	***N***	y	e.g., 64 64 64
ISO 8601	image ID	IMAGE.compression	text	***N***	y	e.g., jpeg
ISO 8601	image ID	IMAGE.dimensionNames	text	***N***	***N***	e.g., spatial, xyz, CTF
ISO 8601	image ID	IMAGE.dimensionOrdering	integer 1D array	***U***	y	e.g., 1 -3 2
ISO 8601	image ID	IMAGE.dimensionRank	integer	***N***	***N***	e.g., 3
ISO 8601	image ID	IMAGE.dimensionSizes	integer 1D array	***N***	***N***	e.g., 3000 3000 3000
ISO 8601	image ID	IMAGE.dimensionUnits	text	***U***	***N***	SI units or pixels
ISO 8601	image ID	IMAGE.pixelDataLocation	float 1D array	***N***	y	e.g., 0.5 0.5 0.5
ISO 8601	image ID	IMAGE.pixelModel	text	***N***	***N***	e.g., float float integer
ISO 8601	image ID	IMAGE.pixelNames	text	***U***	***N***	e.g., density
ISO 8601	image ID	IMAGE.pixels	text	***N***	***N***	URI
ISO 8601	image ID	IMAGE.pixelSizes	float 1D array	***U***	***N***	e.g., .005 .004 .003
ISO 8601	image ID	IMAGE.pixelUnits	text	***U***	***N***	e.g., electrons
ISO 8601	image ID	LINK.xxx	text	y	y	URI
ISO 8601	image ID	LOG.xxx	text	y	y	text string

y=yes, n=no, u=update, xxx=user defined.

Example 1-HDF Organization

description	HDF pathname	comment
imageCore namespace	/imageCore/	
chronological RDF	/imageCore/0	
image pixels	/imageCore/1	original 3D cryo-electron tomogram
optional adjunct group	/imageCore/1*/	
optional metadata	/imageCore/1*/CryoEM.xml	
image pixels	/imageCore/2	subsample of /imageCore/1
image pixels	/imageCore/3	subsample of /imageCore/2

Example 2-HDF Organization integrating non-IC namespaces

description	HDF pathname	comment
imageCore namespace	/imageCore/	
chronological RDF	/imageCore/0	
image pixels	/imageCore/1	linked to /EMAN/Image1
optional adjunct group	/imageCore/1*/	
optional metadata	/imageCore/1*/CryoEM.xml	linked to /EMAN/CryoEM.xml
optional metadata	/imageCore/1*/animation.mpg	
optional metadata	/imageCore/1*/camera.mp7	
image pixels	/imageCore/2	subsample of /imageCore/1
image pixels	/imageCore/3	subsample of /imageCore/2
application namespace	/EMAN/	
application image	/EMAN/Image1	original image
application metadata	/EMAN/CryoEM.xml	linked to /imageCore/3*/CryoEM.xml
application metadata	/EMAN/hdr.ccp4	
application namespace	/Chimera/	
application image	/Chimera/image1/pixels_xyz	linked to /imageCore/1
application image	/Chimera/image1/pixels_xyz_2	linked to /imageCore/2
application image	/Chimera/image1/pixels_xyz_4	linked to /imageCore/3

Please send comments to
Matthew Dougherty, matthewd@bcm.edu

APPENDIX: DISCUSSION

Problems with existing image formats: The vast majority of imaging formats are two dimensional, and therefore are unable to manage n-dimensional complexity. The remaining formats have fixed assumptions tailored to their unique applications, such as 3D spatial, 2D+time, or RGB pixel models; and therefore cannot be adapted. Existing cryo-EM formats are based on outdated file IO methods, such as fread and fwrite; making them unsuitable for very large datasets. For the same reason, existing format designs do not transparently use compression, regional image subsets, and hetero-dataset integration. Generally, they are un-extensible in situations outside their intended application and have sub-optimal IO performance.

Design philosophy: The creation of IC has been made after extended review of imaging designs and recommended best practices of various scientific, computational and archival communities; most notably existing cryo-EM formats, the International Virtual Observatory Alliance, National Information Standards Organization, and the Open Archives Initiative. Several key requirements emerged, specifically the necessity for simplicity, operational high performance, ability to create and interact with large heterogeneous datasets, extensibility, incorporation of existing international standards when appropriate, public processes, open source software, and the need to establish community consensus.

Large Image Data presents two difficult problems. First is the sheer file size that must be processed through the disk drive IO bottleneck, reading the entire image can take considerable time. Second is a caching problem, when the image on a disk drive far exceeds random access memory, the default method is virtual memory management; in this approach the entire image must be memory mapped, requiring a disk file duplication of the image in a transient operating system format. Three methods can be used to better manage these problems: chunking, compression and multi-scale; none which are employed in existing cryo-EM formats. Chunking regionally dissects a dataset into more manageable parts that can be randomly accessed. Compression is useful to reduce the file size and minimize IO transfer time. Multi-scale, or multi-resolution creates sub-sampled images that are useful in establishing regions of interest that can scale to local high resolution, obviating the need to read the entire image.

Pixel Model: The current 3D cryo-EM pixel model is a single scalar value representing density. What is needed is a more robust design that will allow for future developments, such as improved sensors or analytical data. Also a limited pixel model produces a operational barrier for uniformly integrating other non-EM communities' images.

Open Source: There is a general consensus in scientific academic communities that open source software is preferable because of its accessibility and transparency. This becomes an imperative when the scientific data products are destined for public repositories, but there are exceptions such as Microsoft Word or Adobe PDF documents, whose use are ubiquitous and convertible to other formats.

Text: XML and RDF can utilize Unicode; for simplicity IC uses a restricted case of ASCII characters for RDF predicates. It would be desirable to make some of the specification multilingual through the use of Unicode object values, thereby allowing future scientists to make more accurate observations in their native languages. In order to implement this may require the direct use of XML as objects in the case of names and descriptions.

HDF is a unique technology suite that makes possible the management of extremely large and complex data collections. It has been developed over a 20-year period through the support of NCSA, NSF, and NASA. The HDF5 technology suite is designed to organize, store, discover, access, analyze, share, and preserve diverse, complex data in continuously evolving heterogeneous computing and storage environments. The HDF5 data model, file format, API, library, and tools are open and distributed without charge. HDF5 supports all types of data stored digitally, regardless of origin or size. Petabytes of remote sensing data collected by satellites, terabytes of beamline datatsets, and megabytes of high-resolution MRI brain scans are stored in HDF5 files, together with metadata necessary for efficient data sharing, processing, visualization, and archiving. The HDF Group provides a unique suite of technologies and supporting services that make possible the management of large and complex data collections. Its mission is to advance and support HDF technologies and ensure long-term access to HDF data.

XML: Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML (ISO 8879). XML is playing an increasingly important role in the textual exchange of a wide variety of metadata**.** XML is well suited for communities to define and exchange complex metadata due to its self-describing design, but is poorly suited for manipulating large binary datasets that require parallel or random access. To deal with this problem there have been several data frameworks that have sought a balanced integration of XML and HDF in order to benefit from the strong features of each. XML is a product of the World Wide Consortium (W3C) and is hosted by the Massachusetts Institute of Technology Laboratory for Computer Science in the United States, at the European Research Consortium for Informatics and Mathematics in Sophia-Antipolis in France, and at the Keio University Shonan Fujisawa Campus in Japan.

RDF: The Resource Description Framework (RDF) integrates a variety of applications from library catalogs and world-wide directories to syndication and aggregation of news, software, and content to personal collections of music, photos, and events using XML as an interchange syntax. The RDF specifications provide a lightweight ontology system to support the exchange of knowledge on the Web. Because of its simplicity and extensibility it was chosen as the primary means to centrally associate image metadata. RDF is a product of the W3C.

Integrated format specifications and API: The HDFgroup recommends that an application-programming interface accompanies the format specifications. This has several beneficial results: it enforces adherence, provides better testing, enhances compatibility, eases the burden on application programmers, produces an API reference frame based on familiar community norms, and masks the complexities of HDF.

Performance Testing is an absolute necessity because of the underlying requirement of high performance. It will be essential to establish a suite of tests that mirror real world scenarios. Such tests will include simulated datasets and actual operational datasets. Statistical metrics concerning entropy and compression will have to be established. Comparisons across platforms and networks will be needed.

Verification Testing is usually desirable for any complex specification. Software is needed that can examine an IC file and determine compliance, providing detailed notification of non-compliance. Generally, compliance software must examine if the 1) IC-RDF is properly formed, 2) IC-RDF IMAGE.rrr objects properly associate with the underlying HDF constructs, and 3) IC-RDF LINK.xxx objects properly associate to specified metadata.

IMAGE.rrr, such that rrr indicates reserved tags defined by the IC specification. There are no user defined tags.

LINK.xxx, such that xxx indicates user extensible tags; some tags have been reserved. The RDF object is a URI pointing to metadata.

LOG.xxx, such that xxx indicates user extensible tags; some tags have been reserved. The RDF object is a text string.

IMAGE.chunking is an optional predicate that describes the size of chunks used by HDF5; its value is set when the image is created. Chunking regionally subsets a dataset, such as a tile for 2D or sub-volume for 3D. Chunking is typically used in combination with compression. Its size is equal to IMAGE.dimensionRank.

IMAGE.compression is an optional predicate that describes the compression and version used by HDF; its value is set when the image is created. HDF5 currently supports only GZIP and SZIP compression. However, additional compression can be added easily.

IMAGE.dimensionNames describes the axis names corresponding to IMAGE.dimensionSizes. If the names are undefined, the default is ‘pixels’ for each component. The sequence is based on IMAGE.dimensionOrdering default ordering.

IMAGE.dimensionOrdering is an optional predicate that notes the recommended change in the dimension ordering of pixels. Sometimes images are created with unknown symmetry; later additional information (e.g., PDB) can conclusively establish the correct symmetry. For example, a left-handed dataset can noted as right-handed pixel set by setting the IMAGE.dimensionOrdering to {1,2,-3}, or the XY image planes can be diagonally flipped by setting the IMAGE.dimensionOrdering as {2,1,3}. It will be necessary to have an API that will allow arbitrary order in which pixel sequences are extracted from the HDF disk file into computer RAM can be manipulated. The default IMAGE.dimensionOrdering is {1, …, n}, such that n is equal to IMAGE.dimensionRank.

IMAGE.dimensionRank describes the rank of the image and size of related Image.rrr arrays.

IMAGE.dimensionSizes describes the size of each dimension. Its size is equal to IMAGE.dimensionRank. The sequence is based on IMAGE.dimensionOrdering default ordering.

IMAGE.dimensionUnits describes the axis unit values corresponding to IMAGE.dimensionSizes. If the names are undefined, the default is 'pixels' for each component. The sequence is based on IMAGE.dimensionOrdering default ordering.

IMAGE.pixelDataLocation defines the location of data relative to pixel space. A typical assumption is in the center, but the vertex or the face of a picture element has scientific applications. Assuming the 3D pixel is orthorhombic ($\alpha=\beta=\gamma=90°$) implies the data could be located at one of 8 vertices, or on one of 6 faces, or within any point in the pixel space. This matter is important in finite element analysis, segmentation, docking of datasets, and the visualization of datasets. The default IMAGE.pixelDataLocation is {0.5, …, 0.5}, such its size is equal to IMAGE.dimensionRank.

IMAGE.pixelModel is an optional predicate that describes the HDF datatypes classes that are used by the image, except for the compound complex limitation imposed by current version of Pytables.

IMAGE.pixelNames describes the data names corresponding to IMAGE.pixelModel. If the names are undefined, the default is 'pixels' for each component.

IMAGE.pixels describes the original HDF location of the image's pixels, which will be in the IC namespace or user namespace.

IMAGE.pixelSizes describes the unit size of a pixel based on IMAGE.dimensionOrdering default ordering. Its size is equal to IMAGE.dimensionRank.

IMAGE.pixelUnits describes the data units corresponding to IMAGE.pixelModel. If the units are undefined, the default is 'pixels' for each component.

LINK.animation.xxx links to MPEG 1, 2, & 4 animations to an image; xxx indicates user extensible tags.

LINK.application.xxx links application-specific metadata to an image; xxx indicates user extensible tags.

LINK.header.xxx allows one to re-create a specific simple image xxx format by combining the unaltered 'xxx' data format header with the image dataset. Customarily, 'xxx' is the filename extension of the resulting combination.

LINK.mask.xxx links an image that can segment another image, most simply using an AND operator with an image that has the same rank and dimensions. This predicate & related datasets require further development, particularly in regards to multiscale and image-subsets; xxx indicates user extensible tags.

LINK.mime.xxx links MIME metadata to an image; xxx indicates user extensible tags.

LINK.mpeg7.xxx links application-specific metadata to an image; xxx indicates user extensible tags.

LINK.multiscale.xxx links images of different resolution to an image; xxx indicates user extensible tags.

LINK.transformation.xxx links image transformations to image datasets. Image transformations and corresponding metadata transformations can become extremely complex. Regardless of the RDF object values of the RDF predicate 'IMAGE.dimensionOrdering', the initial coordinate assumption of an image is that the first pixel in computer RAM is located at the origin and the remainder of the image is in non-negative space.

'LINK.transformation.base' is reserved for preferred orientations, such as 'LINK.transformation.base_3f'.

Significantly, pixel ordering is computationally simple and coordinate orientation changes can be computationally intense; therefore 90-degree rotations can be simply made changing the dimension order. Further, 3D graphics hardware is fundamentally designed for XYZ right-handed coordinate systems; being able to assemble pixel datasets in various contiguous organizations is crucial for performance, particularly to solve the proper coupling of experimental datasets and visualization tools. Also, there are a variety of scientifically desirable coordinate transformation systems and methods, such as Cartesian, polar, Fourier, Euler angles, quaternion, cosine matrices, and orientation preceding translation; as well as unique reference frames such as world, camera, and object.

Multiple sequential or compound parallel coordinate transformations are not commutative, and several methods are frequently possible to achieve identical results, which is usually left to the imagination of the scientist to select these paths. Therefore it is critical that each step of a transformation pipeline be documented. This diversity will require some form of registry system that contains frequently used transforms and also allows users to dynamically create new transforms and log them with the image.

LOG.citation describes published citations utilizing the image.

LOG.compliance describes software & data compatibility. The RDF object would contain sufficient information to describe 1) what standard and software that image was written by (e.g., 'imageCore v1, IC-API v1.3, EMAN2 v2.1'), and 2) whether the image is read-compatible with other image formats (e.g., 'Chimera map v1').