

The biosciences need an image format capable of high performance and long-term maintenance. Is HDF5 the answer?

BY MATTHEW T. DOUGHERTY, MICHAEL J. FOLK, EREZ ZADOK,
HERBERT J. BERNSTEIN, FRANCES C. BERNSTEIN,
KEVIN W. ELICEIRI, WERNER BENDER, CHRISTOPH BEST

Unifying Biological Image Formats with HDF5

THE BIOLOGICAL SCIENCES need a generic image format suitable for long-term storage and capable of handling very large images. Images convey profound ideas in biology, bridging across disciplines. Digital imagery began 50 years ago as an obscure technical phenomenon. Now it is an indispensable computational tool. It has produced a variety of incompatible image file formats, most of which are already obsolete.

Several factors are forcing the obsolescence: rapid increases in the number of pixels per image;

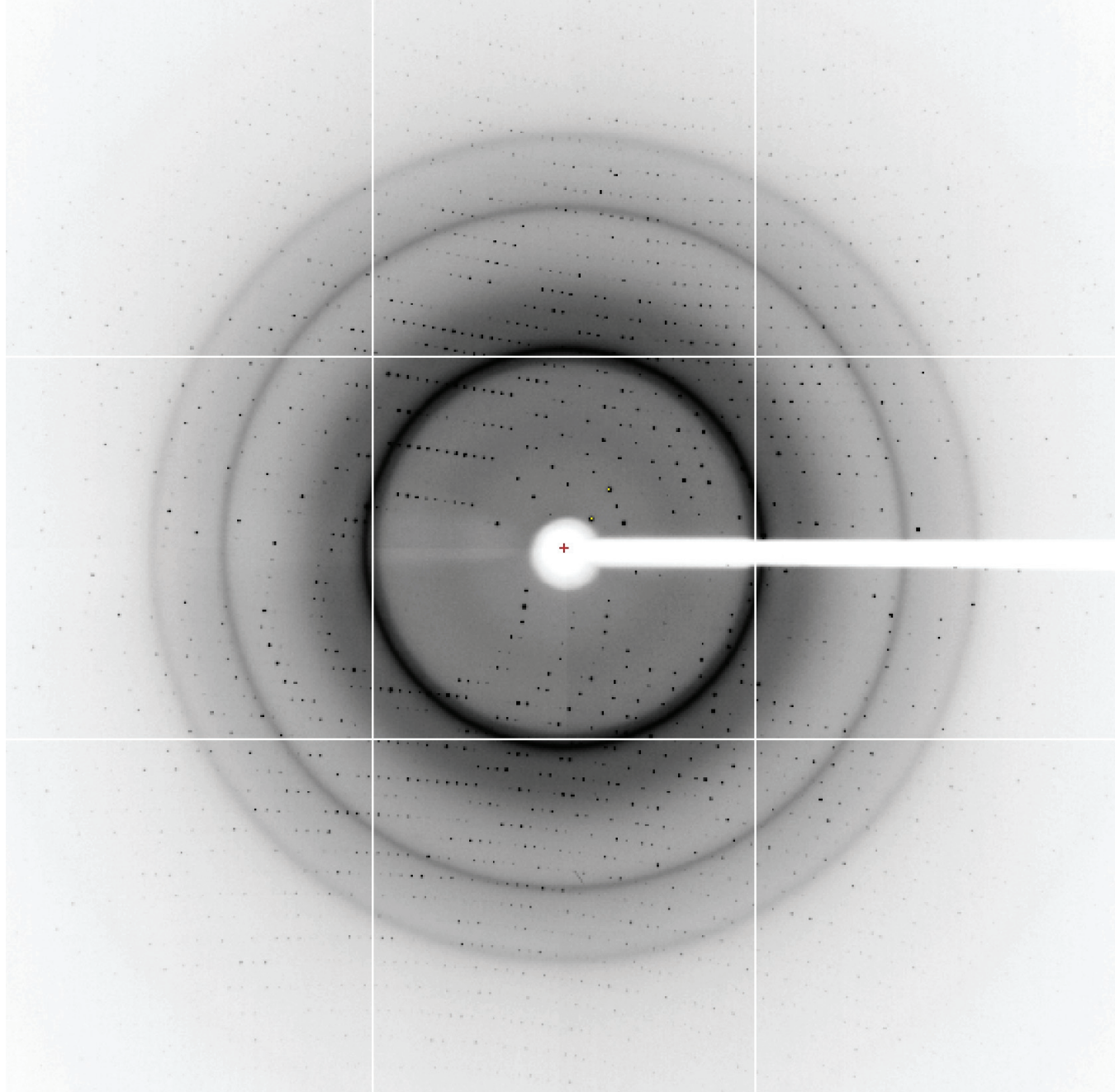
acceleration in the rate at which images are produced; changes in image designs to cope with new scientific instrumentation and concepts; collaborative requirements for interoperability of images collected in different labs on different instruments; and research metadata dictionaries that must support frequent and rapid extensions. These problems are not unique to the biosciences. Lack of image standardization is a source of delay, confusion, and errors for many scientific disciplines.

There is a need to bridge biological and scientific disciplines with an image framework capable of high computational performance and interoperability. Suitable for archiving, such a framework must be able to maintain images far into the future. Some frameworks represent partial solutions: a few, such as XML, are primarily suited for interchanging metadata; others, such as CIF (Crystallographic Information Framework),² are primarily suited for the database structures needed for crystallographic data mining; still others, such as DICOM (Digital Imaging and Communications in Medicine),³ are primarily suited for the domain of clinical medical imaging.

What is needed is a common image framework able to interoperate with all of these disciplines, while providing high computational performance. HDF (Hierarchical Data Format)⁶ is such a framework, presenting a historic opportunity to establish a coin of the realm by coordinating the imagery of many biological communities. Overcoming the digital confusion of incoherent bio-imaging formats will result in better science and wider accessibility to knowledge.

Semantics: Formats, Frameworks, and Images

Digital imagery and computer technology serve a number of diverse biological communities with terminology differences that can result in very different perspectives. Consider the word *format*. To the data-storage community the hard-drive format will play a ma-



An x-ray diffraction image taken by Michael Soltis of LSAC on SSRL BL9-2 using an ADSC Q315 detector (SN901).

jor role in the computer performance of a community's image format, and to some extent, they are inseparable. A format can describe a standard, a framework, or a software tool; and formats can exist within other formats.

Image is also a term with several uses. It may refer to transient electrical signals in a CCD (charge-coupled device), a passive dataset on a storage device, a location in RAM, or a data structure written in source code. Another example is *framework*. An image framework might implement an image standard, resulting in image files created by a software-imaging tool. The framework, the standard, the files, and the tool, as in the case of HDF,⁶ may be so interrelated that they represent dif-

ferent facets of the same specification. Because these terms are so ubiquitous and varied due to perspective, we shall use them interchangeably, with the emphasis on the storage and management of pixels throughout their lifetime, from acquisition through archiving.

Hierarchical Data Format Version 5

HDF5 is a generic scientific data format with supporting software. Introduced in 1998, it is the successor to the 1988 version, HDF4. NCSA (National Center for Supercomputing Applications) developed both formats for high-performance management of large heterogeneous scientific data. Designed to move data efficiently between secondary storage and memory,

HDF5 translates across a variety of computing architectures. Through support from NASA (National Aeronautics and Space Administration), NSF (National Science Foundation), DOE (Department of Energy), and others, HDF5 continues to support international research. The HDF Group, a nonprofit spin-off from the University of Illinois, manages HDF5, reinforcing the long-term business commitment to maintain the format for purposes of archiving and performance.

Because an HDF5 file can contain almost any collection of data entities in a single file, it has become the format of choice for organizing heterogeneous collections consisting of very large and complex datasets. HDF5 is

used for some of the largest scientific data collections, such as the NASA Earth Observation System's petabyte repository of earth science data. In 2008, netCDF (network Common Data Form)¹⁰ began using HDF5, bringing in the atmospheric and climate communities. HDF5 also supports the neutron and X-ray communities for instrument data acquisition. Recently, MATLAB implemented HDF5 as its primary storage format. Soon HDF5 will formally be adopted by the International Organization for Standardization (ISO), as part of specification 10303 (STEP, Standard for the Exchange of Product model data). Also of note is the creation of BioHDF¹ for organizing rapidly growing genomics data volumes.

The HDF Group's digital preservation efforts make HDF5 well suited for archival tasks. Specifically their involvement with NARA (National Archives and Records Administration), their familiarity with the ISO standard *Reference Model for an Open Archival Information System (OAIS)*,¹³ and the HDF5 implementation of the *Metadata Encoding and Transmission Standard (METS)*⁸ developed by the Digital Library Federation and maintained by the Library of Congress.

Technical Features of HDF5

An HDF5 file is a data container, similar to a file system. Within it, user communities or software applications define their organization of data objects. The basic HDF5 data model is simple, yet extremely versatile in terms of the scope of data that it can store. It contains two primary objects: *groups*, which provide the organizing structures, and *datasets*, which are the basic storage structures. HDF5 groups and datasets may also have *attributes* attached, a third type of data object consisting of small textual or numeric metadata defined by user applications.

An HDF5 dataset is a uniform multidimensional array of elements. The elements might be common data types (for example, integers, floating-point numbers, text strings), *n*-dimensional memory chunks, or user-defined compound data structures consisting of floating-point vectors or an arbitrary bit-length encoding (for example, 97-

bit floating-point number). An HDF5 group is similar to a directory, or folder, in a computer file system. An HDF5 group contains links to groups or datasets, together with supporting meta-data. The organization of an HDF5 file is a directed graph structure in which groups and datasets are nodes, and links are edges. Although the term *HDF* implies a hierarchical structuring, its topology allows for other arrangements such as meshes or rings.

HDF5 is a completely portable file format with no limit on the number or size of data objects in the collection. During I/O operations, HDF5 automatically takes care of data-type differences, such as byte ordering and data-type size. Its software library runs on Linux, Windows, Mac, and most other operating systems and architectures, from laptops to massively parallel systems. HDF5 implements a high-level API with C, C++, Fortran 90, Python, and Java interfaces. It includes many tools for manipulating and viewing HDF5 data, and a wide variety of third-party applications and tools are available.

The design of the HDF5 software provides a rich set of integrated performance features that allow for access-time and storage-space optimizations. For example, it supports efficient extraction of subsets of data, multiscale representation of images, generic dimensionality of datasets, parallel I/O, tiling (2D), bricking (3D), chunking (nD), regional compression, and the flexible management of user metadata that is interoperable with XML. HDF5 transparently manages byte ordering in its detection of hardware. Its software extensibility allows users to insert custom software "filters" between secondary storage and memory; such filters allow for encryption, compression, or image processing. The HDF5 data model, file format, API, library, and tools are open source and distributed without charge.

MEDSBIO

X-ray crystallographers formed MEDSBIO (Consortium for Management of Experimental Data in Structural Biology)⁷ in 2005 to coordinate various research interests. Later the electron⁴ and optical¹⁴ microscopy communities began attending. During the past 10 years, each community considered

HDF5 as a framework to create their independent next-generation image file formats. In the case of the NeXus,¹¹ the format developed by the neutron and synchrotron facilities, HDF5 has been the operational infrastructure in its design since 1998.

Ongoing discussions by MEDSBIO have led to the realization that common computational storage algorithms and formats for managing images would tremendously benefit the X-ray, neutron, electron, and optical acquisition communities. Significantly, the entire biological community would benefit from coherent imagery and better-integrated data models. With four bio-imaging communities concluding that HDF5 is essential to their future image strategy, this is a rare opportunity to establish comprehensive agreements on a common scientific image standard across biological disciplines.

Concerns Identified

The following deficiencies impede the immediate and long-term usefulness of digital images:

► *The increase in pixels caused by improving digital acquisition resolutions, faster acquisition speeds, and expanding user expectations for "more and faster" is unmanageable.* The solution requires technical analysis of the computational infrastructure. The image designer must analyze the context of computer hardware, application software, and the operating-system interactions. This is a moving target monitored over a period of decades. For example, today's biologists use computers having 2GB–16GB of RAM. What method should be used to access a four-dimensional, 1TB image having 30 hyperspectral values per pixel? Virtually all of the current biological image formats organize pixels as 2D XY image planes. A visualization program may require the entire set of pixels read into RAM or virtual memory. This, coupled with poor performance of the mass storage relating to random disk seeks, paging, and memory swaps, effectively makes the image unusable. For a very large image, it is desirable to store it in multiple resolutions (multiscale) allowing interactive access to regions of interest. Visualization software may intensively compute these intermediate data resolutions, later discarded upon

exit from the software.

► *The inflexibility of current biological image file designs prevents them from adapting to future modalities and dimensionality.* Rapid advances in biological instrumentation and computational analysis are leading to complex imagery involving novel physical and statistical pixel specifications.

► *The inability to assemble different communities' imagery into an overarching image model allows for ambiguity in the analysis.* The integration of various coordinate systems can be an impassable obstacle if not properly organized. There is an increasing need to correlate images of different modalities in order to observe spatial continuity from millimeter to angstrom resolutions.

► *The non-archival quality of images undermines their long-term value.* The current designs usually do not provide basic archival features recommended by the Digital Library Federation, nor do they address issues of provenance. Frequently, the documentation of a community image format is incomplete, outdated, or unavailable, thus eroding the ability to interpret the digital artifact properly.

Consensus

It would be desirable to adopt an existing scientific, medical, or computer image format, and simply benefit from the consequences. All image formats have their strengths and weaknesses. They tend to fall into two categories: generic and specialized formats. Generic image formats usually have fixed dimensionality or pixel design. For example, MPEG2⁹ is suitable for many applications as long as it is 2D spatial plus 1D temporal using red-green-blue modality that is lossy compressed for the physiological response of the eye. Alternatively, the specialized image formats suffer the difficulties of the image formats we are already using. For example, DICOM³ (medical imaging standard) and FITS⁵ (astronomical imaging standard,) store their pixels as 2D slices, although DICOM does incorporate MPEG2 for video-based imagery.

The ability to tile (2D), brick (3D), or chunk (nD) is required to access very large images. Although this is conceptually simple, the software is not, and must be tested carefully or risk that

subsequent datasets be corrupted. That risk would be unacceptable for operational software used in data repositories and research. This function and its certification testing are critical features of HDF software that are not readily available in any other format.

Common Objectives

The objectives of these acquisition communities are identical, requiring performance, interoperability, and archiving. There is a real need for the different bio-imaging communities to coordinate within the same HDF5 data file by using identical high-performance methods to manage pixels; avoiding namespace collisions between the biological communities; and adopting the same archival best practices. All of these would benefit downstream communities such as visualization developers and global repositories.

Performance. The design of an image file format and the subsequent organization of stored pixels determine the performance of computation because of various hardware and software datapath bottlenecks. For example, many specialized biological image formats use simple 2D pixel organizations, frequently without the benefit of compression. These 2D pixel organizations are ill suited for very large 3D images such as electron tomograms or 5D optical images. Those bio-imaging files have sizes that are orders of magnitude larger than the RAM of computers. Worse, widening gaps have formed between CPU/memory speeds, persistent storage speeds, and network speeds. These gaps lead to significant delays in processing massive data sets. Any file format for massive data has to account for the complex behavior of software layers, all the way from the application, through middleware, down to operating-systems device drivers. A generic n -dimensional multimodal image format will require new instantiation and infrastructure to implement new types of data buffers and caches to scale large datasets into much smaller RAM; much of this has been resolved within HDF5.

Interoperability. Historically the acquisition communities have defined custom image formats. Downstream communities, such as visualization

and modeling, attempt to implement these formats, forcing the communities to confront design deficiencies. Basic image metadata definitions such as rank, dimension, and modality must be explicitly defined so the downstream communities can easily participate. Different research communities must be able to append new types of metadata to the image, enhancing the imagery as it progresses through the pipeline. Ongoing advances in the acquisition communities will continue to produce new and significant image modalities that feed this image pipeline. Enabling downstream users easily to access pixels and append their community metadata supports interoperability, ultimately leading to fundamental breakthroughs in biology. This is not to suggest that different communities' metadata can be or should be uniformly defined as a single biological metadata schema and ontology in order to achieve an effective image format.

Archiving. Scientific images have a general lack of archival design features. As the sophistication of bio-imagery improves, the demand for the placement of this imagery into long-term global repositories will be greater. This is being done by the Electron Microscopy Databank⁴ in joint development by the National Center for Macromolecular Imaging, the RCSB (Research Collaboratory for Structural Bioinformatics) at Rutgers University, and the European Bioinformatics Institute. Efforts such as the Open Microscopy Environment¹⁴ are also developing bio-image informatics tools for lab-based data sharing and data mining of biological images that also are requiring practical image formats for long-term storage and retrieval. Because of the evolving complexity of bio-imagery and the need to subscribe to archival best practices, an archive-ready image format must be self-describing. That is, there must be sufficient infrastructure within the image file design to properly document its content, context, and structure of the pixels and related community metadata, thereby minimizing the reliance on external documentation for interpretation.

The Inertia of Legacy Software

Implementing a new unified image

format supporting legacy software across the biological disciplines is a Gordian knot. Convincing software developers to make this a high priority is a difficult proposition. Implementation occurring across hundreds of legacy packages and flawlessly fielded in thousands of laboratories is not a trivial task. Ideally, presenting images simultaneously in their legacy formats and in a new advanced format would mitigate the technical, social, and logistical obstacles. However, this must be accomplished without duplicating the pixels in secondary storage.

One proposal is to mount an HDF5 file as a VFS (virtual file system) so that HDF5 groups become directories and HDF5 datasets become regular files. Such a VFS using FUSE (Filesystem-in-User-Space) would execute simultaneously across the user-process space and the operating system space. This hyperspace would manage all HDF-VFS file activity by interpreting, intercepting, and dynamically rearranging legacy image files. A single virtual file presented by the VFS could be composed of several concatenated HDF5 datasets, such as a metadata header dataset and a pixel dataset. Such a VFS file could have multiple simultaneous filenames and legacy formats depending on the virtual folder name that contains it, or the software application attempting to open it.

The design and function of an HDF-VFS has several possibilities. First, non-HDF5 application software could interact transparently with HDF5 files. PDF files, spreadsheets, and MPEGs would be written and read as routine file-system byte streams. Second, this VFS, when combined with transparent on-the-fly compression, would act as an operationally usable compressed tarball. Third, design the VFS with unique features such as interpreting incoming files as image files. Community-based legacy image format filters would rearrange legacy image files. For example, the pixels would be stored as HDF5 datasets in the appropriate dimensionality and modality, and the related metadata would be stored as a separate HDF5 1D byte dataset. When legacy application software opens the legacy image file, the virtual file is dynamically recombined and presented by the VFS to the legacy software in the

same byte order as defined by the legacy image format. The fourth possibility is to endow the VFS with archival and performance analysis tools that could transparently provide those services to legacy application software.

Recommendations

To achieve the goal of an exemplary image design having wide, long-term support, we offer the following recommendations to be considered through a formal standards process:

1. Permit and encourage scientific communities to continually to evolve their own image designs. They know the demands of their disciplines best. Implementing community image formats through HDF5 provides these communities flexible routes to a common image model.

2. Adopt the archival community's recommendations on archive-ready datasets. Engaging the digital preservation community from the onset, rather than as an afterthought, will produce better long-term image designs.

3. Establish a common image model. The specification must be conceptually simple and should merely distinguish the image's pixels from the various metadata. The storage of pixels should be in an appropriate dimensional dataset. The encapsulation of community metadata should be in 1D byte datasets or attributes.

4. The majority of the metadata is uniquely specific to the biological community that designs it. The use of binary or XML is an internal concern of the community creating the image design; however, universal image metadata will overlap across disciplines, such as rank, dimensionality, and pixel modality. Common image nomenclature should be defined to bridge metadata namespace conversions to legacy formats.

5. Use RDF (Resource Description Framework)¹⁵ as the primary mechanism to manage the association of pixel datasets and the community metadata. A Subject-Predicate-Object-Time tuple stored as a dataset can benefit from HDF5's B-tree search features. Such an arrangement provides useful time stamps for provenance and generic logging for administration and performance testing. The definition of RDF predicates and objects should

follow the extensible design strategy used in the organization of NFS (Network File System) version 4 protocol metadata.¹²

6. In some circumstances it will be desirable to define adjuncts to the common image model. An example is MPEG video, where the standardized compression is the overriding reason to store the data as a 1D byte stream rather than decompressing it into the standard image model as a 3D YCbCr pixel dataset. Proprietary image format is another type of adjunct requiring 1D byte encapsulation rather than translating it into the common image model. In this scenario, images are merely flagged as such and routine archiving methods applied.

7. Provide a comprehensively tested software API in lockstep with the image model. Lack of a common API requires each scientific group to develop and test the software tools from scratch or borrow them from others, resulting in not only increased cost for each group, but also increased likelihood of errors and inconsistencies among implementations.

8. Implement HDF5 as a virtual file system. HDF-VFS could interpret incoming legacy image file formats by storing them as pixel datasets and encapsulated metadata. HDF-VFS could also present such a combination of HDF datasets as a single legacy-format image file, byte-stream identical. Such a file system could allow user legacy applications to access and interact with the images through standard file I/O calls, obviating the requirement and burden of legacy software to include, compile, and link HDF5 API libraries in order to access images. The duality of presenting an image as a file and an HDF5 dataset offers a number of intriguing possibilities for managing images and non-image datasets such as spreadsheets or PDF files, or managing provenance without changes to legacy application software.

9. Make the image specification and software API freely accessible and available without charge. Preferably, such software should be available under an open source license that allows a community of software developers to contribute to its development. Charging the individual biological imaging communities and laboratories adds

financial complexity to the pursuit of scientific efforts that are frequently underfunded.

10. Establish methods for verification and performance testing. A critical requirement is the ability to determine compliance. Not having compliance testing significantly weakens the archival value by undermining the reliability and integrity of the image data. Performance testing using prototypical test cases assists in the design process by flagging proposed community image design that will have severe performance problems. Defining baseline test cases will quickly identify software problems in the API.

11. Establish ongoing administrative support. Formal design processes can take considerable time to complete, but some needs—such as technical support, consultation, publishing technical documentation, and managing registration of community image designs—require immediate attention. Establishing a mechanism for imaging communities to register their HDF5 root level groups as community specific data domains will provide an essential cornerstone for image design and avoid namespace collisions with other imaging communities.

12. Examine how other formal standards have evolved. Employ the successful strategies and avoid the pitfalls. Developing strategies and alliances with these standards groups will further strengthen the design and adoption of a scientific image standard.

13. Establishing the correct forum is crucial and will require the guidance of a professional standards organization—or organizations—that perceives the development of such an image standard as part of its mission to serve the public and its membership. Broad consensus and commitment by the scientific, governmental, business, and professional communities is the best and perhaps only way to accomplish this.

Summary


Out of necessity, bioscientists are independently assessing and implementing HDF5, but no overarching group is responsible for establishing a comprehensive bio-imaging format, and there are few best practices to rely on. Thus, there is a real possibility that biolo-


gists will continue with incompatible methods for solving similar problems, such as not having a common image model.

The failure to establish a scalable n -dimensional scientific image standard that is efficient, interoperable, and archival will result in a less-than-optimal research environment and a less-certain future capability for image repositories. The strategic danger of not having a comprehensive scientific image storage framework is the massive generation of unsustainable bio-images. Subsequently, the long-term risks and costs of comfortable inaction will likely be enormous and irreversible.

The challenge for the biosciences is to establish a world-class imaging specification that will endow these indispensable and nonreproducible observations with long-term maintenance and high-performance computational access. The issue is not whether the biosciences will adopt HDF5 as a useful imaging framework—that is already happening—but whether it is time to gather the many separate pieces of the currently highly fragmented patchwork of biological image formats and place them under HDF5 as a common framework. This is the time to unify the imagery of biology, and we encourage readers to contact the authors with their views.

Acknowledgments

This work was funded by the National Center for Research Resources (P41-RR-02250), National Institute of General Medical Sciences (5R01GM079429, Department of Energy (ER64212-1027708-0011962), National Science Foundation (DBI-0610407, CCF-0621463), National Institutes of Health (1R13RR023192-01A1, R03EB008516), The HDF Group R&D Fund, Center for Computation and Technology at Louisiana State University, Louisiana Information Technology Initiative, and NSF/EPS-CoR (EPS-0701491, CyberTools). 

 **Related articles**
on queue.acm.org

Catching disk latency in the act

<http://queue.acm.org/detail.cfm?id=1483106>

Better Scripts, Better Games

<http://queue.acm.org/detail.cfm?id=1483106>

Concurrency's Shysters

http://blogs.sun.com/bmc/entry/concurrency_s_shysters

References

1. BioHDF; <http://www.geospiza.com/research/biohdf/>.
2. Crystallographic Information Framework. International Union of Crystallography; <http://www.iucr.org/resources/cif/>.
3. DICOM (Digital Imaging and Communications in Medicine); <http://medical.nema.org>.
4. EMDDB (Electron Microscopy Data Bank); <http://emdatabank.org/>.
5. FITS (Flexible Image Transport System); <http://fits.gsfc.nasa.gov/>.
6. HDF (Hierarchical Data Format); <http://www.hdfgroup.org>.
7. MEDSBO (Consortium for Management of Experimental Data in Structural Biology); <http://www.medsbo.org>.
8. METS (Metadata Encoding and Transmission Standard); <http://www.loc.gov/standards/mets/>.
9. MPEG (Moving Picture Experts Group); <http://www.chiariglione.org/mpeg/>.
10. netCDF (network Common Data Form); <http://www.unidata.ucar.edu/software/netcdf/>.
11. NeXus (neutron, x-ray and muon science); <http://www.nexusformat.org>.
12. NFS (Network File System); <http://www.ietf.org/rfc/rfc3530.txt>.
13. OAIS (Open Archival Information System); <http://nost.gsfc.nasa.gov/isoas/overview.html>.
14. OME (Open Microscopy Environment); <http://www.openmicroscopy.org/>.
15. RDF (Resource Description Framework); <http://www.w3.org/RDF/>.

Matthew T. Dougherty (matthewd@bcm.edu) is at the National Center for Macromolecular Imaging, specializing in cryo-electron microscopy, visualization, and animation.

Michael J. Folk (mfolk@hdfgroup.org) is president of The HDF Group.

Erez Zadok (ezk@cs.sunysb.edu) is associate professor at Stony Brook University, specializing in computer storage systems performance and design.

Herbert J. Bernstein (yaya@dowling.edu) is professor of computer science at Dowling College, active in the development of IUCr standards.

Frances C. Bernstein (fcbb@bernstein-plus-sons.com) is retired from Brookhaven National Laboratory after 24 years at the Protein Data Bank, active in macromolecular data representation and validation.

Kevin W. Eliceiri (eliceiri@wisc.edu) is director at the Laboratory for Optical and Computational Instrumentation, University of Wisconsin-Madison, active in the development of tools for bio-image informatics.

Werner Benger (werner@cct.lsu.edu) is visualization research scientist at Louisiana State University, specializing in astrophysics and computational fluid dynamics.

Christoph Best (best@ebi.ac.uk) is project leader at the European Bioinformatics Institute, specializing in electron microscopy image informatics.