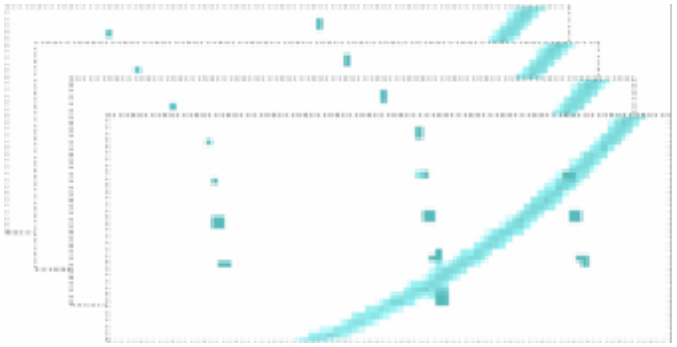
imgCIF, HDF5, NeXus: Issues in Integration of Images from Multiple Sources



Herbert J. Bernstein

Dowling College, Oakdale, NY USA

presentation for

"HDF5 as hyperspectral data analysis format"

Workshop, ESRF, Grenoble, FR, 11-13 January 2010

Work supported in part by grants from U.S. Department of Energy, U.S. NIGMS/NIH and IUCr

Introduction

- When collecting multiple heterogeneous sets data for use in determination of the physical properties of a single target system, heterogeneity of data representations can impede effective analysis. The interaction of imgCIF [Bernstein, Hammersley 2005] , HDF5 [Folk et al. 1999] and NeXus [Klosowski et al. 1998] illuminate important issues in reducing the heterogeneity of the representations so that the essential heterogeneity of the data can be dealt with more effectively and efficiently.
- CIF [Hall, Allen, Brown 1991] [Hall, McMahon 2005] is a table-oriented, i.e. database-oriented, metadata-rich data framework used in crystallography.
- HDF5 is a very general, powerful file-system-like graph-oriented data container used in many data management systems.
- NeXus is a tree-oriented view of HDF5 (and XML and HDF4) of importance in managing neutron and x-ray data.

Why Multiple Formats are Needed

- In crystallography multiple formats are needed. For example, when processing data from the Pilatus 6M pixel array detector [Henrich et al. 2009], in order to keep up with the high data rate, the byte-offset compression in CBFlib [Bernstein, Ellis 2005] is needed. This produces imgCIF format data, while for overall data management of data from multiple experiments, the results may need to be stored in NeXus format.
- In many disciplines one can expect that experiments will require one format for technical issues (performance, unique instrument design, …) and another format for data logging to conform to institutional data management policies and to ensure the preservation of data.

Workshop Goals

Bring together scientists and software developers to find solutions to data format issues from the data analysis point of view,

Put in common different algorithms for analyzing hyper-spectral data to see how they could be extended to different techniques,

Find the most suitable ways to exchange data based on the type of analysis to be performed,

Do hands-on coding for the implementation and/or testing of discussed approaches, ranging from the data format to the analysis algorithms.

The aim of the workshop is not to define how data should be stored at acquisition time or for long-term archival. However, there are several overlapping issues and at least the first morning of the workshop will most likely be dedicated to present and discuss HDF5 and NeXus.

Discussion Points

Choices of data formats influence:

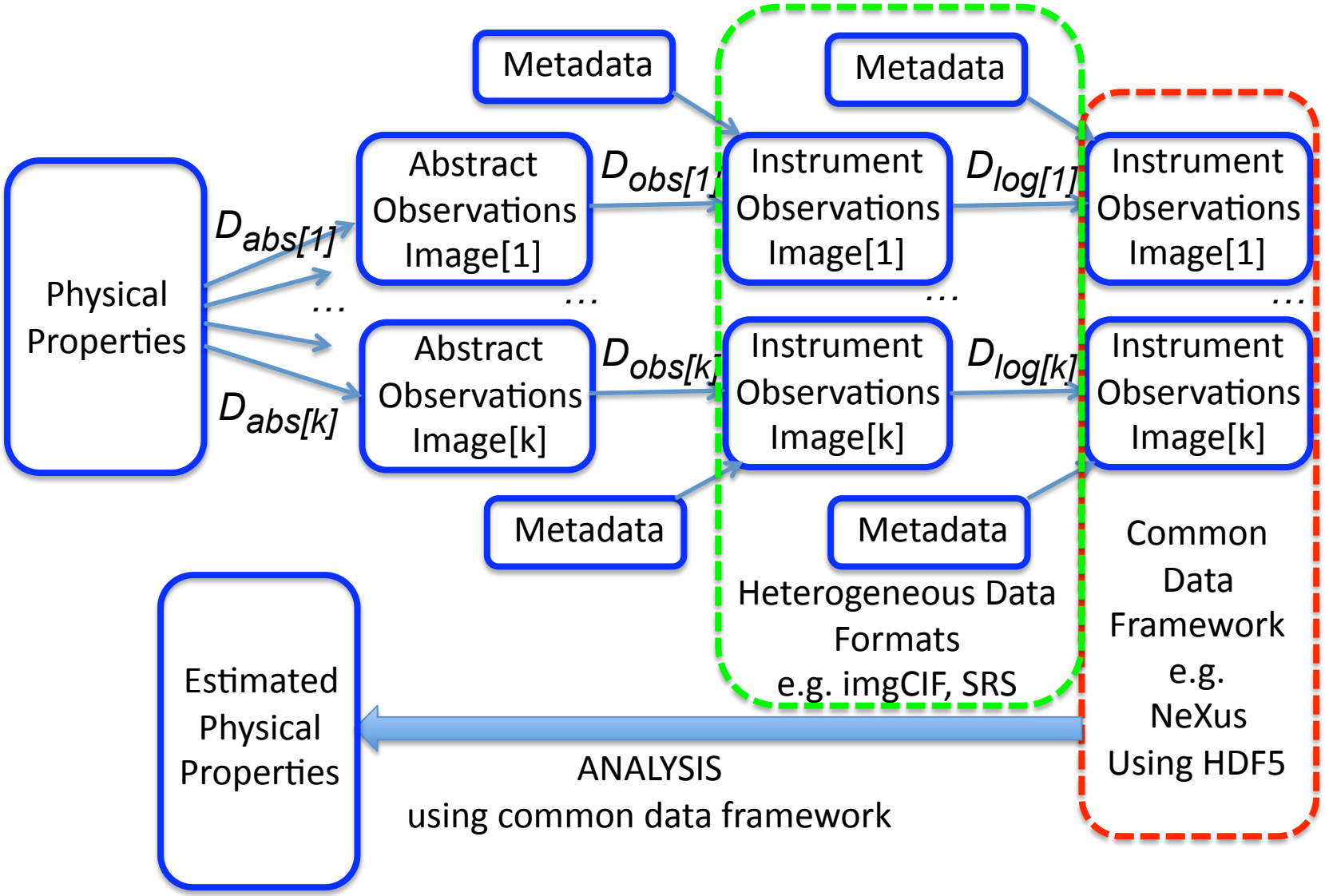
- the accuracy of data analysis
- the design of algorithms for data analysis
- the efficiency and effectiveness of data exchange
  - within a single experiment at one site
  - among multiple experiments at one site
  - within a single experiment at multiple sites
  - among multiple experiments at multiple sites

especially when dealing with multiple sources of data

imgCIF, NeXus, HDF5, XML, … provide useful formats and frameworks. Agreement on what to use is needed.

Suggest a modular approach similar to Kim Henrick's CCP4 Data Harvesting [Henrick 1998]

Suggested Modular Approach



Multiple Sources of Data

Hyperspectral Data: Images or spectra taken in a large number of adjacent narrow spectral bands

Multispectral Data: Images or spectra taken in multiple spectral bands

Multimodal Data: Images or spectra taken by multiple experimental techniques (e.g. diffraction, EM, NMR)

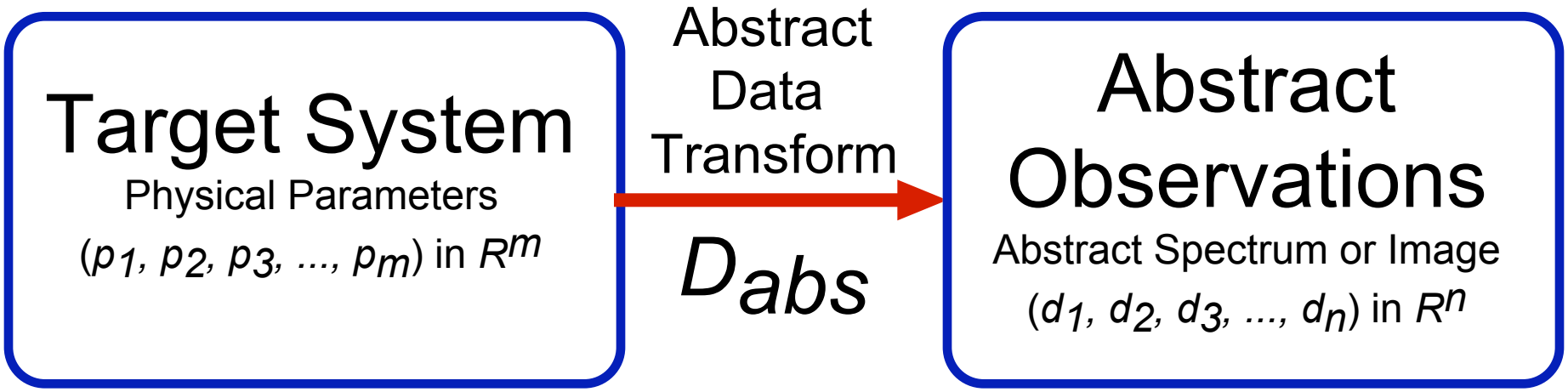
Multisample Data: Images or spectra of different samples of the same experimental target

Formats' Influence on Accuracy

- The choice of data formats has an impact on the achievable accuracy of data analysis.
- The transformations used to convert data to particular formats must be inverted to perform an analysis.
- Loss of numeric accuracy, distortions, aggregation, discarded metadata, ..., imposed by a format in logging transforms can interfere with future data analysis.
- A single format is not likely to be adopted.
- Faithful transformations between formats and lossless compressions are important.

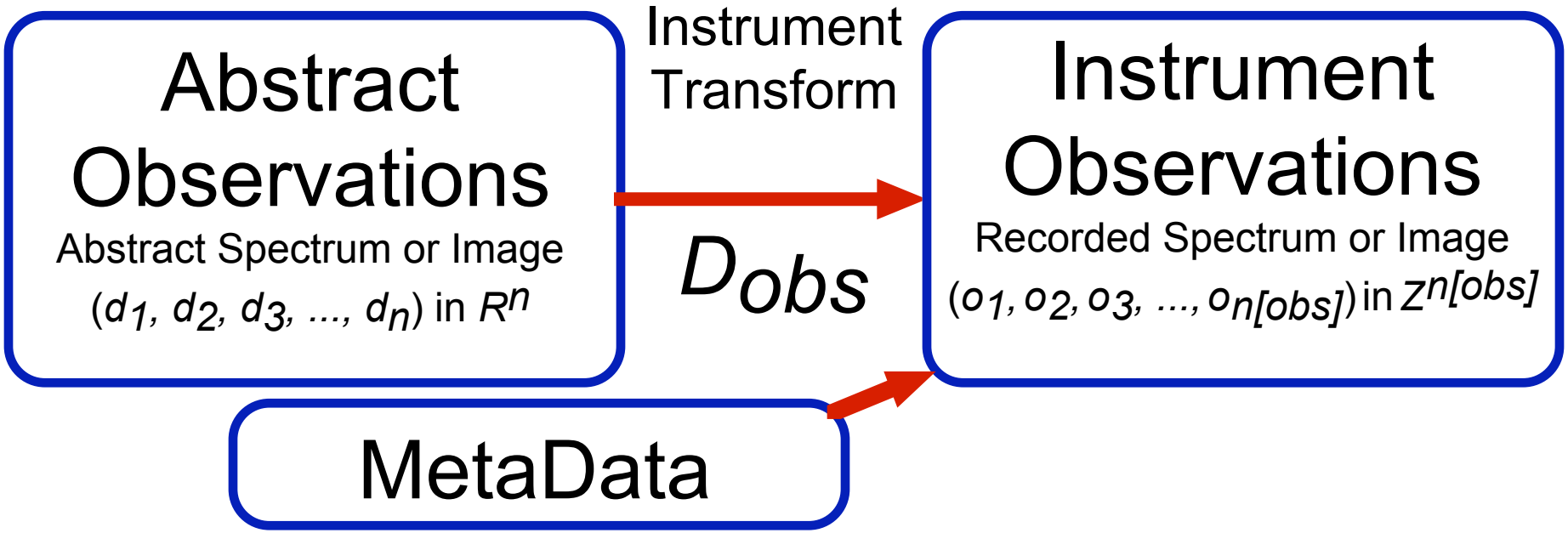
Getting the Data for Analysis

We start with a target system with physical parameters of interest (coordinates, bond lengths, element concentrations, …) and an experimental probe that creates observable spectra or images.



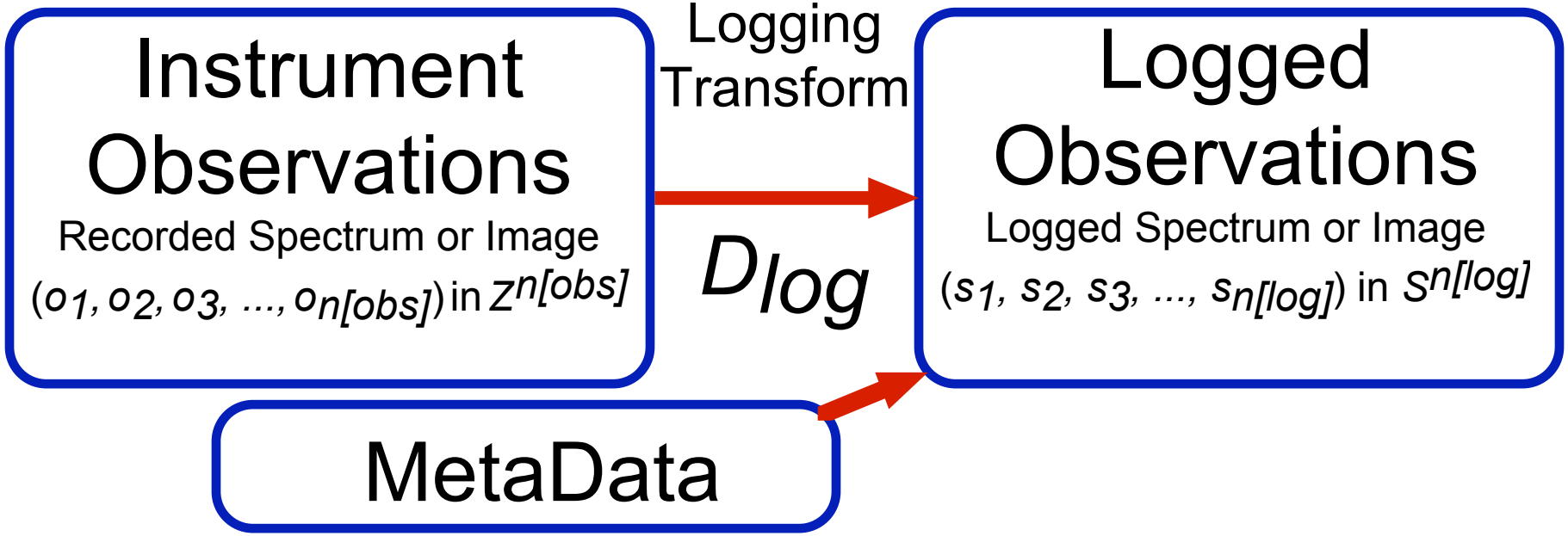
Instrument Transforms

We use instruments to convert abstract observations into measurements that can be recorded. Typically the abstract observations are real numbers or integers with a very wide range. Instruments discretize, aggregate and limit the range of observations and introduce distortions and make other changes to the abstract data and may add further metadata.



Logging Transforms

The data from instruments then is logged in a data management system, which typically adds further metadata about the experiment and which may make changes in the representation of the data, converting numbers to text strings and other bit streams.



Doing Analysis

Ignoring issues of noise, sample degradation, model error, etc., the analysis of logged data is then a matter of inverting the transforms that have been applied to the physical parameters, going from

$$Log = D_{log}(D_{inst}(D_{abs}(Target), \text{Metadata}), \text{Metadata})$$

to

$$Target_{est} = D_{abs}^{-1}(D_{inst}^{-1}(D_{log}^{-1}(Log)))$$

For large systems the number of parameters in the target is likely to be much larger than the number of parameters in the logged data, making the system underdetermined.

Combining Multiple Datasets

Noise is often dealt with by multiple data collections with the same or similar target systems, the same experimental probes and transforms, but for a truly underdetermined system, results from fundamentally different experiments may need to be combined:

$$Log = (Log_1, Log_2, Log_3, \ldots Log_k)$$

$$= (D_{log[1]}(D_{inst[1]}(D_{abs[1]}(Target), Metadata), Metadata),$$
$$D_{log[2]}(D_{inst[2]}(D_{abs[2]}(Target), Metadata), Metadata),$$
$$\ldots)$$

The more carefully we understand and control the differences among the transformation functions used, the better our chances of reliably inverting the composite data log.

Influence on Algorithms

- The choice of data formats has a fundamental impact on the design of algorithms for data analysis.
- Inappropriate data structures can slow access to data and limit the achievable ranges of values.
- Lack of specification of methods in data formats can result in the creation of inconsistent and non-interoperable algorithms.
- Database algorithms require support for key-indexed tables.
- Many algorithms require support for dynamically extensible storage, linked lists and trees.
- For high data-rate applications at synchrotrons, computers and networks impose serious performance limitations, e.g. when working with pixel array detectors.

Support for Algorithms in HDF5

- HDF5 supports arbitrarily complex directed graphs of
  - Groups
  - Datasets
  - Named Datatypes
- An HDF5 Group is similar to a file system directory
- An HDF5 Dataset is similar both to a file in a file system and to an array of data with associated metadata
- An HDF5 Named Datatype is similar to a C struct, giving a named data structure
- One group is distinguished as the root node of the graph.
- Arbitrary links are permitted, but sticking to trees is a more prudent data management approach.

Algorithms as Metadata

- Failure to include algorithms with results may make it difficult to reproduce those results.
- The algorithms for inversion of multiple complex transforms can be very sensitive to seemingly minor changes, e.g. in recent tests of a 3D rigid body fitting algorithm, a change from double (64 bit) to long double (128 bit arithmetic in a vector averaging loop produced changes in the second digit of the rmsd of the fit.
- Algorithms are essential metadata to keep with data in combining multiple datasets.
- Common approach: each data format specifies an API for each language – **this is inefficient and error prone**.

Support for Algorithms in imgCIF

- ImgCIF is being upgraded to DDLm (Dictionary Definition Language with methods [Hall et al. 2008]), part of IUCr CIF2 effort, in progress.
- Methods will be recorded in the dictionaries
  - More efficient than recording each algorithm with each data set
  - But implies a separation of metadata from data
  - And requires access to multiple files
- For more efficient and flexible access to algorithms associated with data, e.g. for visualization scripts, the SBEVSL project [Bernstein, Craig 2006] is working towards allowing DDLm scripts directly in data files as well as in dictionaries once CIF2 is released.

The Basics of imgCIF

There are multiple types of CIF

- DDL1 CIFs (e.g. coreCIF, pdCIF)
- DDL2 CIFs (e.g. mmCIF, imgCIF)
- DDLm and CIF2 are coming

CIF Dictionaries define the terms that can be used and their relationships.

Users can add terms of their own, but one should not use an existing term with a meaning that conflicts with the meaning in a dictionary or in a way that could be confused with terms that have been officially adopted.

For all CIFs:

Information is organized into blocks of data
Each block of data is managed essentially in terms of tables
- Tables are called “categories” or “loops”
- The column headings are called tags” or “data names”
- Some tables have only one row of data
  - then each tag can be put with its value
- Some tables have multiple rows of data

A given tag can appear only once in a block

DDL1 CIFs treat all categories similarly
DDL2 CIFs explicitly state relationships
- e.g. parent-child relationships

imgCIF is a DDL2 dictionary that extends the macromolecular CIF (mmCIF) dictionary.

imgCIF Categories

ARRAY_DATA
- presents the actual numeric data
- (e.g. the numeric values of the pixels in an image)

ARRAY_INTENSITIES
- tells you what you need to do to recover
- intensities form ARRAY_DATA values

ARRAY_STRUCTURE
- how the bits and bytes are organized

ARRAY_STRUCTURE_LIST
- how the array dimensions are organized

ARRAY_STRUCTURE_LIST_AXIS
- how axis settings relate to array indices

AXIS
- the physical parameters of each axis

imgCIF Categories (cont.)

DIFFRN

mmCIF category describing diffraction data

DIFFRN_DATA_FRAME

details about each frame of data

DIFFRN_DETECTOR

information about each detector

DIFFRN_DETECTOR_AXIS

information about each detector axis

DIFFRN_DETECTOR_ELEMENT

layout of detector elements

DIFFRN_MEASUREMENT

goniometer information

DIFFRN_MEASUREMENT_AXIS

information about each goniometer axis

imgCIF Categories (cont.)

DIFFRN_RADIATION
incident radiation (crossfire, polarization, etc.)
DIFFRN_REFLN
reflection-by-reflection parameters for each frame
DIFFRN_SCAN
relationship of axis settings to scans
DIFFRN_SCAN_FRAME
relationship of particular frames to scans
DIFFRN_SCAN_FRAME_AXIS
relationship of axis settings to particular frames

imgCIF Categories (cont.)

Categories under development

MAP

density maps and masks

MAP_SEGMENT

bricks, slices and other segments of maps

Similar categories for compressed binary arrays are being considered.

CIF Syntax

A collection of data blocks

Each data block contains data names (tags) and their values

White space delimits tokens

Tags start with a leading underscore ("_") to distinguish them from values

- Values that might be confused with data names or keywords or that contain whitespace are quoted

Quoting

- single quote (single line only)
- double quote (single line only)
- semicolon in column 1 (multiple lines OK)

- terminal quote mark must be followed by whitespace

Characters with special meaning

- Underscore
- Quote marks
- Period (“.”) or question mark (“?”) (null value)
- Hash mark (“#”) (comment)

Reserved words

- "global_", "data_", "loop_", "stop_", and "save_"

In addition to the underscore, and the three quote marks, three other characters have special meaning: the period ("."), the question mark ("?") and the hash mark ("#"). The period is used when no value is specified. The question mark is used when a value is desired but not available. The hash mark indicates that the remaining characters on a line are part of a comment.

There are a small number of reserved words:
"global_", "data_", "loop_", "stop_", and "save_".
The last two reserved words are not used by CIF but are reserved to prevent conflict with the language from which CIF is derived (STAR).
"global_" and "data_" mark the start of a data block.

"data_" should be followed immediately with the name of the block, without intervening whitespace.

If "loop_" appears, it is followed by a sequence of tags without intervening data values. Those tags are considered as the column headings of a table. These are followed by rows of data values corresponding to those column headings.

Outside of a table, tags and data values appear in simple alternation. Within a data block a given tag may appear only once.

The meaning of a CIF document is not altered by changing the order of presentation of data blocks nor is it altered by changing the order of presentation of tags within a block.

There are two styles of CIF in use for crystallography: DDL1 and DDL2.

DDL1 CIF (e.g. coreCIF, pdCIF)

Partial example of a small molecule coordinate list [Longridge 98]

```
loop_
_atom_site_label
_atom_site_fract_x
_atom_site_fract_y
_atom_site_fract_z
_atom_site_U_iso_or_equiv
_atom_site_adp_type
_atom_site_calc_flag
_atom_site_refinement_flags
_atom_site_occupancy
_atom_site_disorder_assembly
_atom_site_disorder_group
_atom_site_type_symbol
Fe1 1 0 1 .0084(2) Uani d S 1 . . Fe
Na1 .50907(11) .13980(8) 1.09450(9) .0185(3) Uani d . 1 . . Na
Na2 .89904(10) .37128(8) 1.21657(9) .0171(3) Uani d . 1 . . Na
C1 .7997(2) -.01740(18) 1.0419(2) .0110(4) Uani d . 1 . . C
N1 .6788(2) -.02885(18) 1.0696(2) .0166(4) Uani d . 1 . . N
C2 .9306(3) -.01004(16) .8075(3) .0130(4) Uani d . 1 . . C
```

DDL2 CIF (e.g. mmCIF, imgCIF)

Partial example of a macromolecular CIF (1CRN) as converted to mmCIF by the program pdb2cif [Bernstein et al. 98]

```
loop_
_atom_site.label_seq_id
_atom_site.group_PDB
_atom_site.type_symbol
_atom_site.label_atom_id
_atom_site.label_comp_id
_atom_site.label_asym_id
_atom_site.auth_seq_id
_atom_site.label_alt_id
_atom_site.cartn_x
_atom_site.cartn_y
_atom_site.cartn_z
_atom_site.occupancy
_atom_site.B_iso_or_equiv
_atom_site.footnote_id
_atom_site.label_entity_id
_atom_site.id
1 ATOM N N THR * 1 . 17.047 14.099 3.625 1.00 13.79 . 1 1
1 ATOM C CA THR * 1 . 16.967 12.784 4.338 1.00 10.80 . 1 2
```

ImgCIF Binary Data

```
_array_structure.id ARRAY1
_array_structure.encoding_type "signed 32-bit integer"
_array_structure.compression_type packed
_array_structure.byte_order little_endian
_array_data.array_id ARRAY1
_array_data.binary_id 1
_array_data.data
;
--CIF-BINARY-FORMAT-SECTION--
Content-Type: application/octet-stream;
conversions="x-CBF_PACKED"
Content-Transfer-Encoding: BINARY
X-Binary-Size: 3745758
X-Binary-ID: 1
X-Binary-Element-Type: "signed 32-bit integer"
Content-MD5: 1zsJjWPfol2GYl2V+QSXrw==
’ ∏P«q «q FA• f¡Æ• àR˜u<ˇ›k2˙b |5ß …
```

How to Make Changes to the imgCIF Dictionary

1. Get the best current version of the dictionary from the IUCr

2. Check that what you propose is not already there, or if there is at least an appropriate category

3. To avoid conflicts with others doing the same thing, get a prefix from Brian McMahon (bm@iucr.org)

4. If you are going to be sending files to other people, discuss your new definition with them and, please, on the imgcif-l list

5. If this will remain just a local change, use it in good health

6. If you think this should be added to the main dictionary for community use, please say so on the imgcif-l list, and, if appropriate, on other lists

7. If there is sentiment to add it to the main imgCIF dictionary, we will post a revised dictionary for comments, and then, if the dictionary working group agrees, forward the dictionary to COMCIFS for adoption

How to Use and Make or Propose Changes to CBFlib

Use:

1. Download the package (source or binary)
2. If source, build for your machine
3. If you need help building, contact yaya@dowling.edu
4. If you are using the utilities, install them in your favorite location for binaries and use them
5. If you are building an application against the API, install the library in your favorite location and use it

Changes:

1. Changes in your own programs that just use the API:
   Just do it (LGPL)
2. Changes to the API or Program
   Do it, but follow the GPL/LGPL rules on changes
   (making source available, carrying the license forward)

Credit

We would appreciate a credit and knowing about changes.
Please cite [Bernstein, Ellis 2005] (see below)

Nexus

NeXus [Klosowski et al. 1998].

> “NeXus is a data format for the exchange of neutron and synchrotron scattering data between facilities and user institutions. It has been developed by an international team of scientists and computer programmers from neutron and X-ray facilities around the world. The NeXus format uses the hierarchical data format (HDF) that is portable, binary, extensible and self-describing. The NeXus format defines the structure and contents of these HDF files in order to facilitate the visualization and analysis of neutron and X-ray data. In addition, an application program interface (API) [was] produced in order to simplify the reading and writing of NeXus files. The details of the format are available at http://www.neutron.anl.gov/NeXus/”.

NeXus Classes and Groups

A NeXus class is like a CIF category
A NeXus group is like a row in a CIF category

NXentry
similar to a CIF data block
- **NXinstrument**
- **NXsource**
- **NXmoderator**
- **NXcrystal**
- **NXdisk_chopper**
- **NXfermi_chopper**
- **NXvelocity_selector**

NXsample
NXmonitor
NXdata
similar to imgCIF ARRAY_DATA
NXevent_data
NXuser
NXprocess
NXcharacterizations

Integration of imgCIF with NeXus

ImgCIF is tightly structured

- makes it easy to map from imgCIF to other frameworks with less structure
- makes it hard to map from less structured frameworks to imgCIF

Matches need to be found between

- NeXus groups and data items and
- imgCIF categories and tags
  - not just for the items, but for their relationships

Utility cbf2nx available in nexus-4.2

- each data block is converted to an NXentry with the prefix Nxcif with the rest of the CBF tree inserted below it.

Integration with NeXus (cont.)

Handling binary

- No problem with HDF version of NeXus
- Use binUTF to embed binaries in XML version of NeXus

Going from NeXus to imgCIF

- Need to flatten the NeXus hierarchy to 2 levels
- Map each NeXus class to a new CIF category with an “NX_” prefix
- Add explicit tags pointing to parent categories to link the hierarchy together
- Move attributes into tables

For large images, space is an important issue

Image Compression

Compression saves space:

Raw ADSC Quantum 315 detector image:	18.9 MB
imgCIF internally compressed image:	5.8 MB
NeXus HDF5 image (no compression):	37.9 MB
NeXus HDF5 image (LZW compression):	9.4 MB
NeXus raw XML image:	147.2 MB

Important to use HDF5 compression with chunking, but it is important to choose the right compression

Compression takes time. Sometimes weaker compression is necessary to keep up with data flows.

Why not just use the NeXus Tree?

For most purposes storage in a tree is very useful, but when multiple datasets need to be integrated it is best to treat the experimental data as a database.

In order to manage a database reliably, it is best to avoid the explicit pointers used in trees and graphs, and to organize the information in tables linked implicitly by key values in each row.

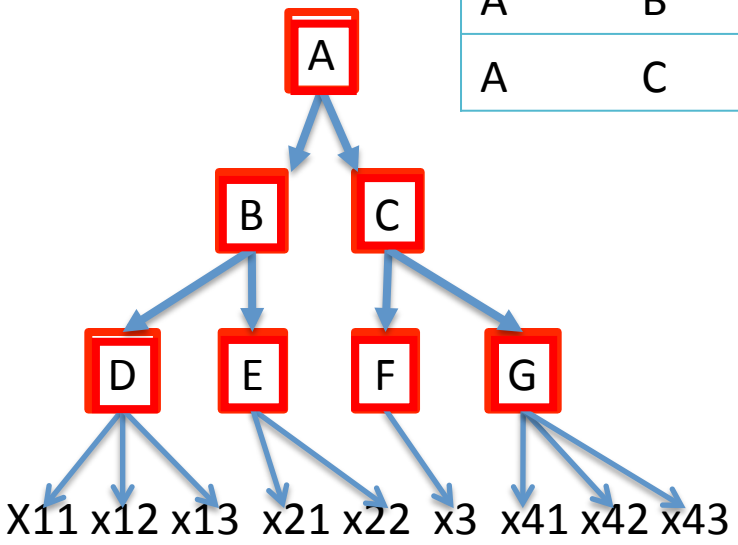
The conversion from a tree to these “relational” tables is called normalization and is the basis of all modern database implementations.

How do you map a graph into tables?

Each non-leaf node in the graph becomes a table

Each leaf node become an entry in the table

Each link in the graph becomes a node column link column pair



Node	Link
A	B
A	C

Node	Link
B	D
B	E

Node	Link
C	F
C	G

Node	Data
D	x11
D	x12
D	x13

Node	Data
E	x21
E	x22

Node	Data
G	x41
G	x42
G	x43

Node	Data
F	x3

The resulting tables can be combined

Conclusion

- Multiple formats are a reality of multisource data.
- Multiple formats are often desirable.
- But multiple formats can inhibit reliable data analysis, introducing errors and delays and loss of data in translation.
- Hub formats capable of efficient flexible faithful representation of data without loss of metadata are desirable.
- imgCIF, HDF5 and NeXus can all contribute to a workable framework for management of data in multiple formats.

References

[Bernstein, Ellis 2005] Bernstein, H. J., Ellis, P. J. (2005). “CBFlib: An ANSI C Library for Manipulating Image Data”, chapter 5.6 in “International Tables For Crystallography, Volume G: Definition and Exchange of Crystallographic Data,” S. R. Hall and B. McMahon, eds., International Union of Crystallography, Springer, Dordrecht, NL, pp. 544 – 556.

[Bernstein, Craig 2006] Bernstein, H. J., Craig, P. A. (2006) “Structural Biology Extensible Visualization Scripting Language”, http://sbevsl.sourceforge.net/

[Bernstein, Hammersley 2005] Bernstein, H. J., Hammersley, A. P. (2005) “Specification of the Crystallographic Binary File (CBF/imgCIF)”, chapter 2.3 in “International Tables For Crystallography, Volume G: Definition and Exchange of Crystallographic Data,” S. R. Hall and B. McMahon, eds., International Union of Crystallography, Springer, Dordrecht, NL, pp. 37 – 43.

[Codd 1970] Codd, E. F. (1970) “A Relational Model of Data for Large Shared Data Banks”, Comm. ACM 13:6 pp. 377 – 387.

[Folk et al. 1999] Folk, M., Cheng, A., McGrath, R. E. (1999) “HDF5: A New File Format and I/O Library for Scientific Data Management”, Astronomical Data Analysis software and Systems VIII Proceedings, David M. Mehringer, Raymond L. Plante, and Douglas A. Roberts, eds., Astronomical Society of the Pacific, Volume 172, 1999.

[Hall, Allen, Brown 1991] Hall, S. R., Allen, F. H. & Brown, I. D. (1991), ‘The crystallographic information file (CIF): a new standard archive file. for crystallography’, Acta Cryst. A47, 655 – 685.

References (cont.)

[Hall, McMahon 2005] Hall, S. R. & McMahon, B. (2005), Volume G: Definition and Exchange of Crystallographic Data, International Tables For Crystallography, Springer: Dordrecht, chapter 1.1. Genesis of the Crystallographic Information File.

[Hall et al. 2008] Hall, S. Spadiccini, Westbrook, J. (2008) "DDLm: Next Generation Dictionary Definition Language", (Version: 13 August 2008), http://journals.chester.iucr.org/iucr-top/cif/ddlm/DDLm_13aug08/DOC/DDLm_spec_aug08.pdf

[Henrich et al. 2009] Henrich, B., Bergamaschi, A, Broennimann, C., Dinapoli, R., Eikenberry, E. F., Johnson, I., Kobas, M., Kraft, P., Mozzanica, A., Schmitt, B (2009) "PILATUS: a single photon counting pixel detector for X-ray applications", Nuclear Inst. and Methods in Physics Research, A, 607:1, pp. 247 – 249.

[Henrick 1998] Henrick, K. (1998) "CCP4 and Data Harvesting," CCP4 Newsletter on Protein Crystallography, No. 35, July 1998.

[Klosowski et al. 1998] Klosowski, P.; Koennecke, M.; Tischler, J. Z.; Osborn, R. (1998). "NeXus: A common format for the exchange of neutron and synchrotron data", *Physica B: Physics of Condensed Matter*, 241,1-4, pp. 151-153.

Software status

CBFlib:

http://www.bernstein-plus-sons.com/CBF
http://sourceforge.net/projects/cbflib

API (C function library, under GPL or LGPL, your choice)
more compressions, major speedup, support for maps
Manual
Sample files
Utilities

SVN repository on sourceforge

NeXus:

http://www.nexusformat.org

SBEVSL:

http://sbevsl.sourceforge.net
http://blondie.dowling.edu/projects/sbevsl

Acknowledgements

The organizers of the “HDF5 as hyperspectral data analysis format” for their hospitality

U.S. DOE, U.S. NIH, IUCr, U.S. NSF for current and past research funding

Some of the people, present* and past, who have been involved:

Frances C. Bernstein*

ARCiB Lab at Dowling College: HJB*, Isaac Awuah Asiamah, Darina Boycheva, Georgi Darakev, Nikolay Darakev, Jonathan Ihm*, John Jemilawon, Nan Jia, Petko Kamburov, Gregory McQuillan, Daniel O’Brien, Matt Rousseau*, Georgi Todorov and Elena Zlateva*

Paul Craig’s Group at RIT: PAC* , Eno Akpovwa, Abdul Bangura, Anthony Corbett, Luticha Doucette, Chanelle Francis, Brett Hanson, Katrina Henry, MaryEd Kenney, Desiree Matthews, Scott Mottarella, Mario Rosa*, Charlie Westin, Corey Wischmeyer

Additional Material

Additional material to help in understanding imgCIF is provided on the slides that follow.

Status of CIF in PX

Core CIF (used for ligands):

an effective, working standard, heavily used

mmCIF (macromolecular structures):

some use, community prefers old PDB format.

imgCIF (synchrotron data images, other images):

used for Pilatus 6M detector

supported by major detector vendors for compliance

some general use starting

Problem

Use of many different data formats causes delay and confusion; may lead to errors

Synchrotron image data formats:

Wladek Minor is “dealing with 197 (!!!!) frame formats” [email Wladek Minor to H. J. Bernstein, 15 May 2006].

2003 Denzo manual lists 107 available detector formats [Gewirth 2003] [Otwinowski, Minor 1997]

An Impractical Solution
(The Esperanto Solution)

Mandate one perfect format for everyone to use for everything (internals and externals)

Not workable:

Might suppress new ideas and good science

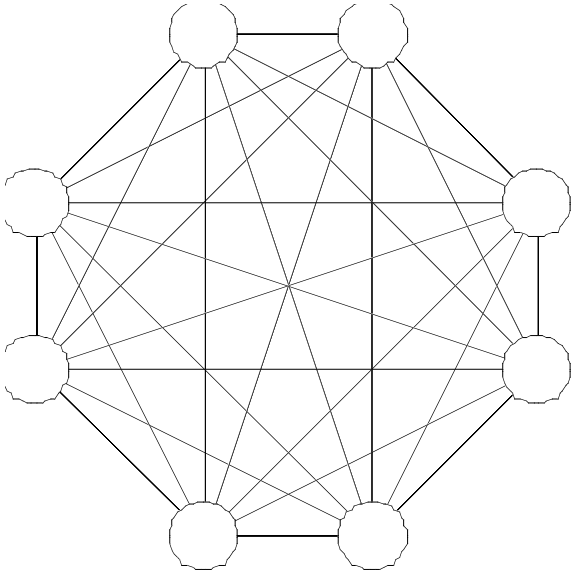
There is no agreement on the “perfect” format

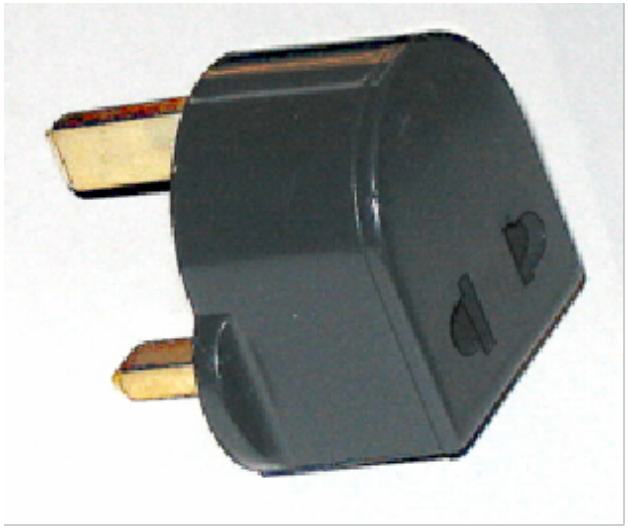
What is perfect for the internals of one project might be imperfect for the internals of another project

A Solution: Focus on Interchange

Include adapters with/in each system to handle many formats

For N systems with N-1 foreign formats, need N*(N-1) adapters



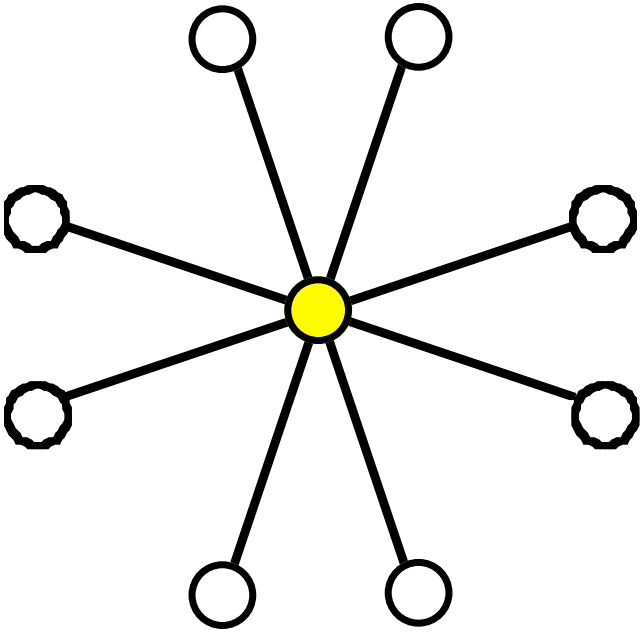


A Better Solution

Provide a common format for interchange (e.g. CIF, NeXus)

Instead of N*(N-1) conversions

We only need 2N conversions



Background

Imagine trying to do electronics without any agreement as to how to draw schematics.

That is how we have managed massive amounts of data in structural biology.

Some points of coherence in the chaos:

- BNL Protein Data Bank format [Bernstein et al. 1977]
  - *de facto* standard for macromolecular coordinates
- CCDC's Cambridge Structural Database [Allen et al. 1973]
  - *de facto* standard for small molecules

Background (cont.)

In general

Each vendor of data collection equipment

Each group maintaining a software package

has their own approach to representing and storing raw data

This causes delays and confusion -- friction -- in working with data.

Is the Problem Real?

Date: Sat, 06 Jan 2007 16:32:12 -0500

From: Arun Malhotra <malhotra@miami.edu>

To: CCP4 Bulletin Board <ccp4bb@dl.ac.uk>

"One common source of errors is changes made in reflection or coordinate files… For example, only recently, someone in my lab just manually edited out a hkl format file to change a few reflections from the exponential format into the standard F format, so that it could be converted into the mtz format. … Format conversions (hkl to mtz or vice-versa) or simple manual edits to coordinate files are very common, and are fertile places for mistakes to creep in. Once such mistakes are made, they are not often easy to catch since there is no easy way to compare files…."

Is the Problem Real?

Date: Sun, 07 Jan 2007 11:25:26 +0900

From: Charlie Bond <Charles.Bond@uwa.edu.au>

To: ccp4bb@dl.ac.uk

"… Often at the synchrotron one is in a tired hurry to get an image indexed and processed. If the wrong parameters are used (eg the ones from the home lab with a bit of editing), a dataset in the wrong hand can be quickly produced

"Increasingly beamlines automatically prepare the correct parameter files for you, but it is cases where images are difficult to process (low resolution, disorder) that processing may occur later at home and the details of the beamline may be disregarded.

"Correct me if I'm wrong, but even the deposition of images would not help this as the critical information is the geometry of the beamline set up which is probably not recor[d]ed with the images.

…"

Frequently Asked Questions

Can I change imgCIF?

Yes, please do. We would appreciate:

- New ideas
- New items for the dictionary
- New support software
- Bug fixes and improvements
- Translations to and from other presentations

Please don't use existing terms in ways that conflict with their meanings; define a new term with a new name instead.

Frequently Asked Questions (Cont.)

The BIG Frequently Asked Question

Can I make proprietary software using imgCIF and CBFlib?

Yes, the API in CBFlib is available under the LGPL.

If you change CBFlib itself, you must publish the changed source code under the LGPL, but even if you change CBFlib, you do not have to make your program into an open source program.

Where to Find imgCIF Information

IUCr Crystallographic Information Framework:

International Tables, Volume G

http://www.iucr.org/iucr-top/cif/index.html

official copies of dictionaries and stable releases of software

Image CIF/Crystallographic Binary File (imgCIF/CBF)

http://arcib.dowling.edu/CBF

http://www.bernstein-plus-sons.com/software/CBF

development versions of dictionary and software

http://www.iucr.org/iucr-top/cif/cbf/imgcif-l

http://scripts.iucr.org/mailman/listinfo/imgcif-l

imgCIF discussion list (please join)

Management of Experimental Data in Structural Biology (MEDSBIO)

http://www.medsbio.org

A broader perspective (imgCIF, NeXus, …) concentrating on interfaces

http://www.medsbio.org/meetings

information on this workshop and future ones of interest

http://scripts.iucr.org/pipermail/medsbio-l/

http://scripts.iucr.org/mailman/listinfo/medsbio-l

MEDSBIO discussion list (please join)

Protein Data Bank

http://www.pdb.org

Information on dictionaries and file format, BioSync, etc.