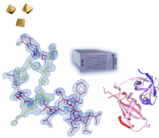




Use of the CBF Library in the Context of Real-Time Image Analysis

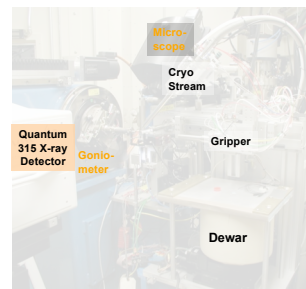
Nicholas Sauter | Lawrence Berkeley National Lab

Collaborator: Ana González | Stanford Synchrotron Radiation Lab



Standardization of the crystallographic image format has the potential for making data processing software much more portable. LABELIT, the LBNL autoindexing toolbox, currently supports data formats produced by most equipment vendors, but the many *ad hoc* rules required to implement this support make the code difficult to maintain. The present goal is to utilize the CBF library for image processing at the earliest possible date. CBF-formatted images will be treated within a data processing pipeline, such as that offered by Stanford's *Web-Ice* package. This allows image characteristics to be immediately analyzed and reviewed, to optimize the experimental protocol.

Experimentation at Modern Automated Beamlines



Eventual Goals:

- Fully automated data collection with multi-wavelength protocol

Present Goals:

- Screen for best crystal growth conditions
- Select the highest-quality samples from a batch
- Discovery of drug leads and protein-ligand complexes
- Enable multi-crystal dataset acquisition
- Perform initial characterization with minimal radiation dose

- >100 Crystal Samples; liquid nitrogen autofill
- All hutch equipment under motorized control

Reliable* Software for High-Throughput Automation



DISTL Zhang *et al.* (2006) *J Appl Cryst* 39:112

- Immediate identification of candidate Bragg spots (< 4 seconds)
- Pick out artifacts like ice rings
- Estimate resolution limits

LABELIT Sauter *et al.* (2004) *J Appl Cryst* 37:399

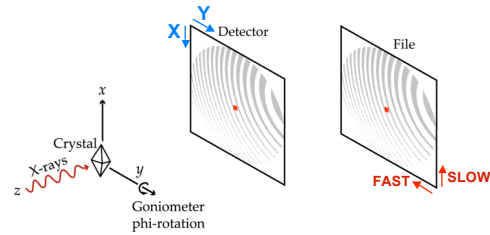
- Fast characterization of lattice without recourse to graphical interface (< 20 seconds)
- Estimate mosaicity
- Prepare input file for automatic spot integration with *MOSFLM*

Web-Ice González *et al.* (2005) *Acta Cryst A* 61:C486

- Remote (Web) access to immediate results
- Tabular view permits comparison of multiple samples
- Calculation of optimal data acquisition settings

* That is, improved. Macromolecular diffraction patterns are very diverse. Basic well-known algorithms (e.g., cell reduction & autoindexing) had to be rewritten to cover outlying cases. Legacy software (pre-2003) relied heavily on human input to recognize the challenging cases.

Inter-Related Coordinate Systems



- At present the coordinate relationships must be worked out individually for each detector type.
- We must guard against writing CBF files that contain no information about either the detector type or the coordinate system transformations.

Support for Multiple Detector Vendors



• Detector Geometry

- Flat surface, square pixels
 - ADSC Quantum 4, 210, 315
 - Mar CCD
 - Mar Image Plate
 - Rigaku Raxis IV and HTC
 - Rigaku Raxis II (after transformation from rectangular shape)
 - Rigaku Saturn 92 CCD
 - MacScience DIP 2030b
- Only limited success with other geometries
 - Bruker Proteus CCD (1K x 1K)
 - APS SBC 19BM / 19ID requires calibration file

Thanks for providing information
Chris Nielson, ADSC
Michael Blum, Mar USA
Jim Pflugrath, Rigaku



Example Difficulties with Existing File Formats



- Local keyword dialects. The openness of the ADSC file format has allowed different facilities to utilize conflicting keywords.
 - Berkeley Center uses conflicting "DENZO_BEAM_CENTER" and "BEAM_CENTER" tags
- Coordinate system relationships are unspecified
 - There are 8 possible relationships between Detector and File coordinate systems. For ADSC detectors, two of them are in common use at different synchrotrons. LABELIT needs to maintain a list, keyed by DETECTOR serial number.
- Unit of measure is unspecified
 - ESRF writes MAR CCD beam center in mm instead of pixel units
- Redundant information
 - Start phi, end phi, and delta phi all defined.

All of these problems can be corrected with CBF

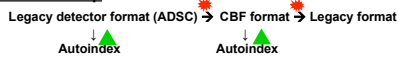


```
# Image CIF Dictionary (imgCIF) #
# and Crystallographic Binary File Dictionary (CBF) #
# Extending the Macromolecular CIF Dictionary (mmcIF) #
# Version 1.4.0A01 #
# of 2006-07-04 #

This example show the axis specification of the axes of a detector, source and gravity.
The order has been changed as a reminder that the ordering of presentation of tokens is not significant.
The centre of rotation of the detector has been taken to be 68 millimetres in the direction away from the source.

source . source . 0 0 1 1 . .
gravity . gravity . 0 -1 0 . .
trans translation detector rotx 0 0 1 0 0 -68
twotheta rotation detector . 1 0 0 . .
rotxy rotation detector twotheta 0 1 0 0 0 -68
rotx rotation detector roty 0 0 1 0 0 -68
```

First validation step



Second validation step



Distribute validation programs and test cases in parallel with libCBF code

Intended Use of CBF within LABELIT



- Link the C-language CBF library to an existing core library of C++ crystallography algorithms, the "Computational Crystallography Toolbox". Grosse-Kunstleve et al. 2002, *J Appl Cryst* 35: 126. (<http://cctbx.sourceforge.net>)
- Encapsulate the complexities of CBF function within wrapper C++ classes, exposing only the limited set of features that will actually be used for file reading and data processing, although this could be extended at any time.
- Abstract CBF data structures into C++ classes so that memory management can be handled by constructors and destructors.
- Re-define the error-handling macros so that C++ exceptions will be thrown and handled by the user code.
- Expose the C++ wrapper classes at the Python scripting level with Boost.Python bindings
- Use Python scripts to rapidly prototype new approaches for data processing.

As work progresses, information for downloading source code and test cases for this sub-project will be posted at <http://cci.lbl.gov/labelit>