DXTBX: the diffraction experiment toolbox

James Parkhurst\textsuperscript{1}, Aaron Brewster\textsuperscript{2}, Luis Fuente\textsuperscript{e}-Montero\textsuperscript{3}, David Waterman\textsuperscript{3}, Johan Hattne\textsuperscript{2}, Nathanial Echols\textsuperscript{2}, Gwyndaf Evans\textsuperscript{1}, Nicholas Sauter\textsuperscript{2}, Graeme Winter\textsuperscript{1}

\textsuperscript{1}Diamond Light Source, \textsuperscript{2}Lawrence Berkeley National Laboratory, \textsuperscript{3}STFC Rutherford Appleton Laboratory

E-mail: james.parkhurst@diamond.ac.uk

Despite a push towards the adoption of industry standards, the recording of X-ray diffraction data remains fragmented. Processing data from single crystal X-ray diffraction experiments, therefore, requires the ability to read, and interpret, image and metadata from a variety of data formats employing different experimental representations. Tools have previously been developed to address this problem. The CCP4 Diffraction Image library \cite{remacle2007} was developed to support the DNA and xia2 projects; however, it was limited by a lack of extensibility. FabIO \cite{knudsen2013} provides generic access to 2D image data but does not give consistent access to experimental models derived from image metadata.

The dxtbx provides a consistent interface to both image data and experimental models whilst supporting a completely generic, user-extensible approach to reading the data and metadata from different sources. Both single image files, such as imgCIF \cite{bernstein2005}, and multiple-image files, such as HDF5, are supported. It was originally developed for use in xia2 \cite{winter2009} to replace the Diffraction Image library; however, it has since been incorporated into the cctbx \cite{grosse-kunstleve2002}. The library is written in a mixture of C++ and Python, is open source and is distributed under a BSD license.

Access to the experimental models and image data is provided through a high-level ‘sweep’ interface instantiated by a factory function from the input data representation. A sweep represents a series of images that have a well defined geometric relationship between adjacent pixels in 3D, e.g. a series of images taken using the rotation method. The sweep provides convenient access to image data through a python list-style interface and provides methods to extract arbitrary 3D volumes from the images sequence. Simple access to experimental geometry is provided through four container classes: beam, detector, goniometer and scan. The detector model, being necessarily the most complex, also provides, for example, methods to predict intersections by diffracted beam vectors; the concept of a virtual detector plane \cite{bricogne1987} is used in order to provide abstraction from the device dependent underlying implementation. Complex effects such as parallax correction and arbitrary pixel alignments are handled transparently through a configurable pixel to millimetre mapping function. In this way, general algorithms can be developed without reference to the specifics of the experimental setup and hardware used.

---

\textsuperscript{1}Remacle, F. and Winter, G., (2007), CCP4 newsletter on protein crystallography.


Keywords: software; experimental geometry; diffraction image data

---

Keywords: software; experimental geometry; diffraction image data