XFEL Detectors and ImageCIF

Aaron S. Brewster[a], Johan Hattne[a], James M. Parkhurst[b], David G. Waterman[c], Herbert J. Bernstein[d], Graeme Winter[b], and Nicholas K. Sauter[a]

[a]*Lawrence Berkeley National Laboratory, Berkeley, CA 94720*
[b]*Diamond Light Source, Harwell Science and Innovation Campus, OX11 0DE (UK)*
[c]*CCP4, Research Complex at Harwell, STFC Rutherford Appleton Laboratory, OX11 0FA (UK)*
[d]*Department of Mathematics and Computer Science, Dowling College, Oakdale, NY 11769*

Correspondence email: asbrewster@lbl.gov

Introduction

Serial femtosecond crystallography performed using X-ray free electrons lasers (XFELs) creates a challenging task for modern detectors (Chapman *et al.* 2011, Boutet *et al.* 2012, Kern *et al.* 2013). Pulses containing 10^{12} photons, 40-50 femtoseconds (fs) long are generated using a linear accelerator and impact a liquid jet of protein microcrystals at a rate of 120 pulses per second. Still image diffraction patterns from thousands of crystals can be collected in a matter of minutes. The Cornell-SLAC pixel array detector (CSPAD) is a unique detector designed to operate at these rates and record data from exposures on the fs time scale (Hart *et al.* 2012). The CSPAD is modularized into 32 sensors arranged in a roughly square pattern. This creates unique challenges for representing the data in such a way that the geometric layout of the experiment is accurately recorded. To this end, we have adopted and extended the ImageCIF/CBF file specification (Bernstein & Hammersley 2005) to record CSPAD diffraction data, adding sufficient parameters to the ImageCIF dictionary to lay out the XFEL experiment at SLAC, including fully specifying the detector geometry. The new ImageCIF parameters described below help us handle the detector geometry by expressing it easily refineable terms. This extensibility and the ability to parameterize the entire experiment explicitly made ImageCIF/CBF the best option for representing this data. The CSPAD CBF format is natively understood by *cbflib* (Ellis & Bernstein 2001), a software package developed specifically for reading and writing CBF files. We have also incorporated the format into *cctbx* (Grosse-Kunstleve *et al.* 2002, Sauter *et al.* 2013), using a new multi-tile detector model defined by the module *dxtbx* (the diffraction experiment toolbox)(Waterman *et al.* 2013, Parkhurst *et al.* in preparation). These software packages allow us to refine the experimental geometry against measured data, leading to better indexing rates and more accurate integration of the reflection signal (Hattne *et al.* submitted).

CSPAD Detector Geometry

Three full-size CSPAD detectors are in service at the present, two at the Linac Coherent Light Source (LCLS) Coherent X-ray Imaging (CXI) instrument, and one at the LCLS X-ray Pump Probe (XPP) instrument. Each detector is comprised of 32 sensors and each sensor houses two application-specific integrated circuits (ASICs), 194x185 pixels in dimension, with a pixel size of 110 microns and a three-pixel gap between them (Figure 1). 8 sensors comprising 16 ASICs form a quadrant. The four quadrants surround a central hole, through which the undiffracted beam passes. The CXI detector quadrants are adjustable on diagonal rails, allowing the central size to grow and shrink. This allows the second detector, typically positioned 2.5 meters behind the first, to receive signal.

The 32 sensors are not orthogonalized, meaning edges between sensors are not parallel; each sensor is tilted slightly off of 90°. Further, the sensors are not co-planar with the detector, having small angles off of the planar normal. LCLS provides optical measurements to position the sensors in three-dimensional space, and these measurements have been enormously useful in specifying the detector geometry. At the CXI beamline, quadrant positions are not provided, as they are variable. Their location needs to be experimentally determined, initially by aligning the quadrants to rings from powder diffraction, and subsequently refined against single crystal diffraction. For both CXI and XPP, detector tilt and position need to be refined as well. For example, the beam itself is not always perfectly parallel to the detector rail, leading to small changes in beam center at different detector distances. All of this geometric information needs to be recorded for each still in a way understandable by developers working on indexing and integration while still

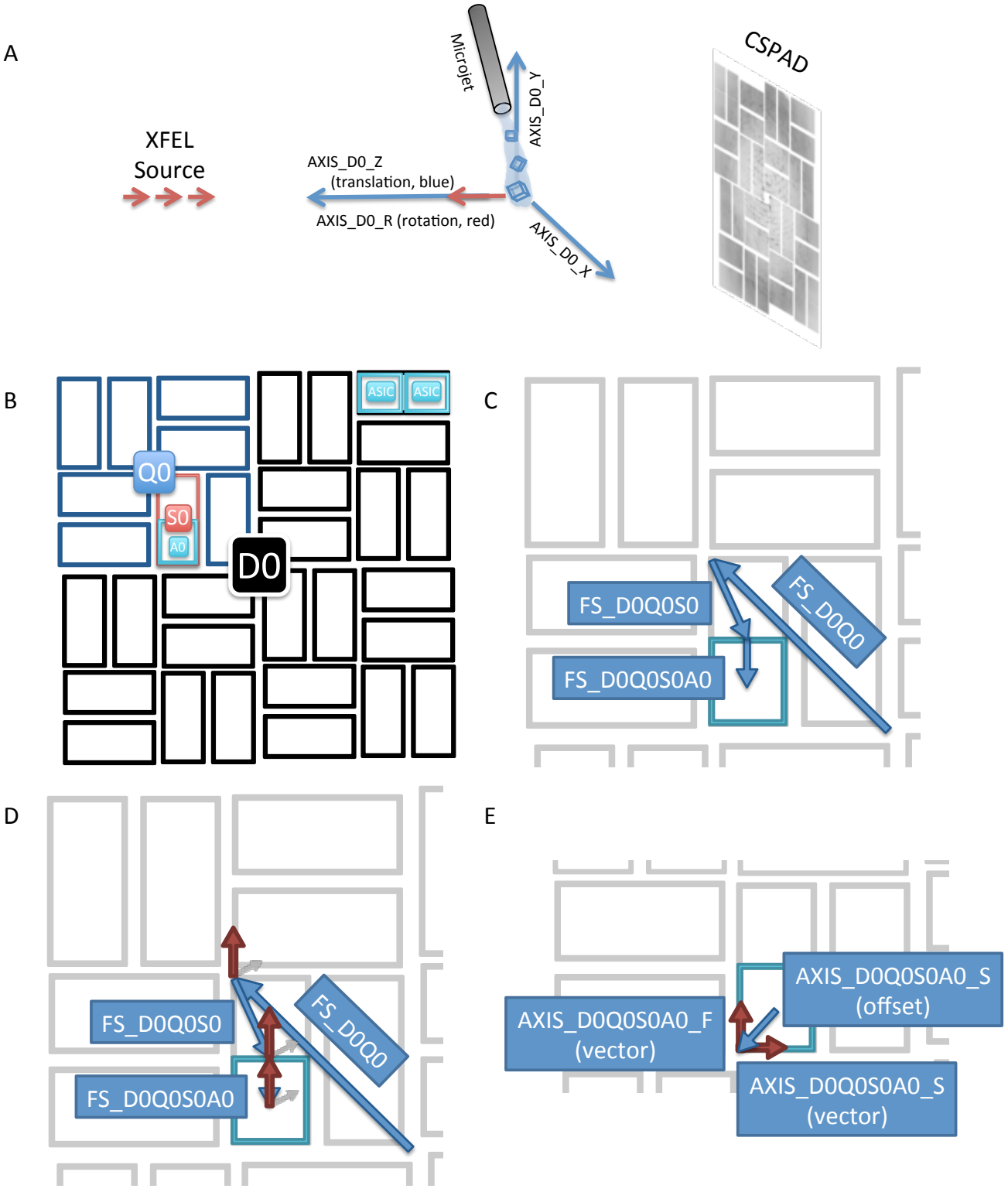

Figure 2: ImageCIF axes describing the CSPAD detector. A) XFEL experiment overview. Crystals are injected into the XFEL stream via a micro-injection system. The root ImageCIF axes for the CSPAD detector as a whole are shown. Axes AXIS_D0_X, _Y and _Z are translation axes along which the detector can be moved. Detector distance is specified as a translation along AXIS_D0_Z. A fourth axis, AXIS_D0_R, defines a rotation axis around which the detector can be rotated. B) Overview of the detector. Rectangles are 32

Caption continues on the following page.

sensors, each comprising 2 ASIC pixel array chips, as shown in the upper right hand corner. D0, Q0, S0 and A0 are highlighted, corresponding to detector zero (*i.e.*, the detector as a whole), quadrant zero in the upper left hand corner, sensor zero and ASIC zero. C) Frame shift vector offsets for the three rotation axes that position the quadrant, sensor, and ASIC centers. CBF rotation axes have three components: an offset from the base of the parent axis to its base, a vector describing the direction around which to rotate, and a rotation angle. Three axis offsets are shown (blue) that together describe the position of an ASIC relative to the center of the detector. D) Vector components (red) of the three rotation axes shown in B. Arrows are normal to the respective surface planes. Around these axes the various elements are rotated into position by specifying appropriate angles. E) Vectors (red) and offsets (blue) for the 2 fast and slow axes of an ASIC element. Here, the red vectors are translation axes that are co-planar with the ASIC chip. They relate how the pixel array is laid out in space to its in-memory arrangement. The blue arrow is the slow axis' offset from the center of the ASIC. The fast axis depends on the slow axis so its offset is zero.

being easily parameterized for refinement.

ImageCIF/CBF

ImageCIF is a specific CIF dictionary for representing diffraction data. Binary encoding of the pixel array data together with ImageCIF metadata comprises the Crystallographic Binary File format (CBF). In use by a variety of companies to record diffraction frame data, ImageCIF and CBF are internationally agreed upon standards maintained by the International Union of Crystallography. ImageCIF allows complete description of the geometry of the crystallographic experiment. For these reasons, we found it applicable to our needs.

In ImageCIF, one describes frame data in the form of a blueprint for the detector. First, the individual pixel-array elements are defined. In the case of the CSPAD, 64 elements are specified:

```
loop_
_diffrn_detector_element.id
_diffrn_detector_element.detector_id
 ELE_D0Q0S0A0 CSPAD_FRONT
 ELE_D0Q0S0A1 CSPAD_FRONT
 ELE_D0Q0S1A0 CSPAD_FRONT
 ELE_D0Q0S1A1 CSPAD_FRONT
...
 ELE_D0Q3S7A0 CSPAD_FRONT
 ELE_D0Q3S7A1 CSPAD_FRONT
```

Here, a new CIF table is defined with the 'loop_' keyword, named diffrn_detector_element. The table links elements by their IDs to a detector ID (CSPAD_FRONT). Multiple detectors can be defined in the same file; if the second detector at CXI, known as the back detector and positioned up to 2.5 meters behind the front detector, is in use, its data and metrology could be recorded in the same file. The convention we use for naming the CSPAD elements includes IDs for the detector (D), quadrant (Q), sensor (S) and ASIC (A). Thus ELE_D0Q0S1A0 is the array of pixels that represents detector 0, quadrant 0, sensor 1, ASIC 0. Later in the file, each of these elements has a separate binary encoding of their pixel data. Other tables specify gain, array dimensions and further physical properties of each element.

Once elements are defined, the geometry of the detector is laid out using two tables: axis and diffrn_scan_frame_axis. The axis table specifies lines of motion and axes of rotation for the experiment, while the diffrn_scan_frame_axis table specifies physical settings for detector components along the axes specified in the axis table. The ImageCIF axis convention specifies the origin to be the sample position, with the X-axis pointing along the axis of right-handed goniometer rotation, the Z-axis pointing to the beam source, and the Y-axis completing a right handed system (Figure 1A). In the case of many XFEL experiments, no goniometer is present, so the X-axis is simply orthogonal to the beam and gravity. Thus the first few lines of the CSPAD axis table are shown in scheme 1.

Each line defines an axis by its type: general, translation or rotation. Equipment refers to kinds of devices that move along the given axis. Other examples include goniometer axes, which XFEL experiments generally do not include. The vector specifies either the direction of translation or the axis about which rotation is performed, and the offset positions the base of the axis in space relative to the parent axis specified in the depends_on field. Finally, equipment_component is a new field we have added to the ImageCIF dictionary in collaboration with its principal maintainer, Herbert Bernstein. This field allows us to group axes together, which will be important to distinguish hierarchy level later when we construct a detector model using *dxtbx* software.

Axis positions are specified in the diffrn_scan_frame_axis table:

```
loop_
_axis.id
_axis.type
_axis.equipment
_axis.depends_on
_axis.vector[1]
_axis.vector[2]
_axis.vector[3]
_axis.offset[1]
_axis.offset[2]
_axis.offset[3]
_axis.equipment_component
 AXIS_SOURCE     general     source    .              0  0 1   .   . . .
 AXIS_GRAVITY    general     gravity   .              0 -1 0   .   . . .
 AXIS_D0_Z       translation detector  .              0  0 1   .   . . detector_arm
 AXIS_D0_Y       translation detector AXIS_D0_Z       0  1 0   .   . . detector_arm
 AXIS_D0_X       translation detector AXIS_D0_Y       1  0 0   .   . . detector_arm
 AXIS_D0_R       rotation    detector AXIS_D0_X       0  0 1   0   0 0 detector_arm
```

Scheme 1: ImageCIF 'loop' table. The first 12 lines comprise a header in which the table and each of its 11 columns are named. The first 6 axes are also shown, describing the detector as a whole and its orientation in the laboratory.

```
 FS_D0Q0         rotation    detector AXIS_D0_R       0  0 1 -50  42 0 detector_quadrant
 FS_D0Q0S0       rotation    detector FS_D0Q0         0  0 1  11 -23 0 detector_sensor
 FS_D0Q0S0A0     rotation    detector FS_D0Q0S0       0  0 1 -11   0 0 detector_asic
```

Scheme 2: These three axis entries in the loop table correspond to frameshifts describing the transition from the detector as a whole to quadrant 0, from quadrant 0 to sensor 0, and from sensor 0 to ASIC 0.

```
 AXIS_D0Q0S0A0_S translation detector FS_D0Q0S0A0     0 -1 0 -11  10 0 detector_asic
 AXIS_D0Q0S0A0_F translation detector AXIS_D0Q0S0A0_S 1  0 0   0   0 0 detector_asic
```

Scheme 3: The fast and slow axes of an ASIC. The slow axis is offset (-10, 11, 0) mm from the ASIC center and points in the -Y direction (in relation to its parent). The fast axis depends on the slow axis and points in the X direction (in relation to its parent).

```
loop_
_diffrn_scan_frame_axis.axis_id
_diffrn_scan_frame_axis.frame_id
_diffrn_scan_frame_axis.angle
_diffrn_scan_frame_axis.displacement
 AXIS_SOURCE  FRAME1 0    0
 AXIS_GRAVITY FRAME1 0    0
 AXIS_D0_X    FRAME1 0    0
 AXIS_D0_Y    FRAME1 0    0
 AXIS_D0_Z    FRAME1 0 -171
 AXIS_D0_R    FRAME1 0    0
```

While both angle and displacement can be specified, only the one or the other is meaningful, depending on if the axis is a rotation or translation axis. The detector distance is specified by translating 171 mm along AXIS_D0_Z (in the negative Z direction since Z points to the source).

Once the detector position is specified, subcomponents are laid out in the axis table (Figures 1B, 1C and 1D) as shown in scheme 2. Here, the frame shifts needed to position quadrant 0, detector 0 and asic 0 are specified. Because these are not mechanical axes, we adopt a convention of naming them FS_ for frame shift instead of AXIS_. These are rotation axes to allow sensor rotations to be specified in the diffrn_scan_frame_axis table:

```
 FS_D0Q0      FRAME1     0 0
 FS_D0Q0S0    FRAME1 89.7 0
 FS_D0Q0S0A0 FRAME1     0 0
```

We can see that sensor 0 is rotated 89.7 degrees around its rotation axis specified in the axis table, the (0, 0, 1) axis *i.e.* the Z-axis. In reality, the sensor is tilted slightly from normal. Another CBF file we have generated records the sensor 0 axis vector to be (-0.000974376302058 0.00044773585801 0.999999425062), indicating a very slight tilt from the normal (about 0.6°). ImageCIF allows us to record even this small error, improving the accuracy of the detector description.

Finally, the fast and slow axes are specified for each asic tile in the axis table (Figure 1E, scheme 3). Note that the slow axis is offset from the center

```
# read the file
image = reader(filename)

# iterate through the quadrants of the detector
detector = reader.get_detector()

quadrants = detector.hierarchy()
for quadrant in quadrants:
       # vector pointing to the center of the quadrant relative to the
       # center of the detector
       origin = quadrant.get_origin()

       # unit vectors pointing in the fast and slow directions of the
       # quadrant plane
       fast = quadrant.get_fast_axis()
       slow = quadrant.get_slow_axis()

       # these three vectors form a 3D basis for this quadrant
       <optimize 9 parameters against a set of measured data>

       quadrant.set_frame(refined_fast,
                          refined_slow,
                          refined_origin)

# apply the detector object changes to the image's internal cbf handle
image.sync_detector_to_cbf

#write the new file
image._cbf_handle.write_file(new_filename)
```

Scheme 4: Pseudo-Python code describing a possible optimization of the four quadrant positions.

of the ASIC, positioning it at the (0, 0) pixel. The fast and slow axes are unit vectors that specify the readout directions for the data stored in the CBF binary sections. These entries, together with the above information, completely describe the detector geometry.

dxtbx and CSPAD ImageCIF

Recently, we have collaborated with researchers at the Diamond Light Source in the UK to develop a new *cctbx* component, the diffraction experiment toolbox *dxtbx*. This toolbox provides Python and C++ based interfaces for generically reading crystallographic data regardless of file format. Importantly, the toolbox exposes models of the diffraction experiment through a set of four interfaces, the detector, the scan, the goniometer and the beam. The developer can sub-class from more general file reader classes and expose the detector geometry through these interfaces. For the purpose of XFEL data (still data), only the detector and beam models are useful.

We have written an appropriate generic reader for multi-tile detector data in CBF format, and ensured its compatibility with this CSPAD CBF format. The reader reads the axis list and creates a hierarchy of components using the equipment_component tag in the axis table to group axes together. Scheme 4 is an example of Python code that uses this reader to read a CSPAD CBF file and show how the hierarchy can be used to refine quadrant positions.

The hierarchical model provides powerful tools for interacting with detector geometry to accomplish tasks of importance to XFEL data collection in a straightforward manner.

Finally, XFEL sources can produce hundreds of thousands of individual diffraction patterns. Representation of each pattern as a single CBF file in hard disk storage can be detrimental to file system performance, a problem exacerbated when handling large numbers of experimental runs, each with many files. Use of HDF5 reduces the file system burden for large numbers of runs by grouping multiple images into large HDF5 files, reducing the burden for each run. Therefore, optional conversion of CBF/ImageCIF files to HDF5/NeXus in *CBFlib* is under development (Bernstein *et al.* 2013). The hierarchical geometries presented here will be preserved, with

the added benefit that metadata only needs to be recorded once per complete dataset in an single HDF5 master file, as opposed to being repeated in thousands of separate CBF image files, each containing a full header description (see also the *Computational Crystallography Newsletter* companion article in this issue, "Coping with BIG DATA image formats: integration of CBF, NeXus and HDF5").

Conclusion

Integration of XFEL intensity data requires precise knowledge of where individual pixels are in physical space. Spot centroids are used for indexing, followed by crystal unit cell and orientation refinement. Correct refinement will predict spot locations such that integration masks will capture true signal while avoiding background. The ImageCIF/CBF representation we are implementing in *cctbx.xfel* for the CSPAD detector allows for simpler refinement of detector geometry, at the detector, quadrant, sensor and ASIC levels to sub-pixel accuracy. Incorporation into *dxtbx* enables straightforward access to detector and beam models, facilitating this refinement.

Acknowledgements

This work was supported by NIH grants GM095887 and GM102520 and Director, Office of Science, Department of Energy (DOE) under contract DE-AC02-05CH11231 for data-processing methods (N.K.S.).

References

Bernstein HJ and Hammersley AP (2005). Specification of the Crystallographic Binary File (CBF/imgCIF). *International Tables For Crystallography*. H. S. R. and M. B. Dordrecht, NL, Springer. **G:** 37-43.

Bernstein HJ, Sloan JM, Winter G, Richter TS, NeXus International Advisory Committee and Committee on the Maintenance of the CIF Standard (2013). "Coping with BIG DATA Image Formats: Integration of CBF, NeXus and HDF5." *poster, American Crystallographic Association, 2013 Annual Meeting*. Honolulu, HI.

Boutet S, Lomb L, Williams GJ, Barends TR, Aquila A, Doak RB, Weierstall U, DePonte DP, Steinbrener J, Shoeman RL, Messerschmidt M, Barty A, White TA, Kassemeyer S, Kirian RA, Seibert MM, Montanez PA, Kenney C, Herbst R, Hart P, Pines J, Haller G, Gruner SM, Philipp HT, Tate MW, Hromalik M, Koerner LJ, van Bakel N, Morse J, Ghonsalves W, Arnlund D, Bogan MJ, Caleman C, Fromme R, Hampton CY, Hunter MS, Johansson LC, Katona G, Kupitz C, Liang M, Martin AV, Nass K, Redecke L, Stellato F, Timneanu N, Wang D, Zatsepin NA, Schafer D, Defever J, Neutze R, Fromme P, Spence JC, Chapman HN and Schlichting I (2012). "High-resolution protein structure determination by serial femtosecond crystallography." *Science* **337**: 362-364.

Chapman HN, Fromme P, Barty A, White TA, Kirian RA, Aquila A, Hunter MS, Schulz J, DePonte DP, Weierstall U, Doak RB, Maia FR, Martin AV, Schlichting I, Lomb L, Coppola N, Shoeman RL, Epp SW, Hartmann R, Rolles D, Rudenko A, Foucar L, Kimmel N, Weidenspointner G, Holl P, Liang M, Barthelmess M, Caleman C, Boutet S, Bogan MJ, Krzywinski J, Bostedt C, Bajt S, Gumprecht L, Rudek B, Erk B, Schmidt C, Homke A, Reich C, Pietschner D, Struder L, Hauser G, Gorke H, Ullrich J, Herrmann S, Schaller G, Schopper F, Soltau H, Kuhnel KU, Messerschmidt M, Bozek JD, Hau-Riege SP, Frank M, Hampton CY, Sierra RG, Starodub D, Williams GJ, Hajdu J, Timneanu N, Seibert MM, Andreasson J, Rocker A, Jonsson O, Svenda M, Stern S, Nass K, Andritschke R, Schroter CD, Krasniqi F, Bott M, Schmidt KE, Wang X, Grotjohann I, Holton JM, Barends TR, Neutze R, Marchesini S, Fromme R, Schorb S, Rupp D, Adolph M, Gorkhover T, Andersson I, Hirsemann H, Potdevin G, Graafsma H, Nilsson B and Spence JC (2011). "Femtosecond X-ray protein nanocrystallography." *Nature* **470**: 73-77.

Ellis P and Bernstein H (2001). "CBFlib: An API for CBF/imgCIF Crystallographic Binary Files with ASCII Support.

Grosse-Kunstleve RW, Sauter NK, Moriarty NW and Adams PD (2002). "The Computational Crystallography Toolbox: crystallographic algorithms in a reusable software framework." *Journal of applied crystallography* **35**: 126-136.

Hart P, Boutet S, Carini G, Dubrovin M, Duda B, Fritz D, Haller G, Herbst R, Herrmann S, Kenney C, Kurita N, Lemke H, Messerschmidt M, Nordby M, Pines J, Schafer D, Swift M, Weaver M, Williams G, Zhu D, Van Bakel N and Morse J (2012). "The CSPAD megapixel x-ray camera at LCLS." *Proceedings of SPIE* **8504**: 85040C-85040C-85011.

Hattne J, Echols N, Tran R, Kern J, Gildea R, Brewster A, Alonso-Mori R, Glöckner C, Hellmich J, Laksmono H, Sierra R, Lassalle-Kaiser B, Lampe A, Han G, Gul S, DiFiore D, Milathianaki D, Fry A, Miahnahri A, White W, Schafer D, Seibert M, Koglin J, Sokaras D, Weng T, Sellberg J, Latimer M, Glatzel P, Zwart P, Grosse-Kunstleve R, Bogan M, Messerschmidt M, Williams G, Boutet S, Messinger J, Zouni A, Yano J, Bergmann U, Yachandra V, Adams P and Sauter N (submitted). "The accurate processing of diffraction data from X-ray free-electron lasers."

Kern J, Alonso-Mori R, Tran R, Hattne J, Gildea RJ, Echols N, Glockner C, Hellmich J, Laksmono H, Sierra RG, Lassalle-Kaiser B, Koroidov S, Lampe A, Han G, Gul S, Difiore D, Milathianaki D, Fry AR, Miahnahri A, Schafer DW, Messerschmidt M, Seibert MM, Koglin JE, Sokaras D, Weng TC, Sellberg J, Latimer MJ, Grosse-Kunstleve RW, Zwart PH, White WE, Glatzel P, Adams PD, Bogan MJ, Williams GJ, Boutet S, Messinger J, Zouni A, Sauter NK, Yachandra VK, Bergmann U and Yano J (2013). "Simultaneous femtosecond X-ray spectroscopy and diffraction of photosystem II at room temperature." *Science* **340**: 491-495.

Parkhurst J, Brewster A, Fuentes-Montero F, Waterman D, Hattne J, Ashton A, Echols N, Evans G, Sauter N and Winter G (in preparation). "dxtbx: the diraction experiment toolbox."

Sauter NK, Hattne J, Grosse-Kunstleve RW and Echols N (2013). "New Python-based methods for data processing." *Acta crystallographica. Section D, Biological crystallography* **69**: 1274-1282.

Waterman DG, Winter G, Parkhurst JM, Fuentes-Montero L, Hattne J, Brewster A, Sauter NK and Evans G (2013). "The DIALS framework for integration software." *CCP4 Newsletter on Protein Crystallography* **49**: 13-15.