A Gold Standard for Macromolecular Diffraction Data

Andreas Förster,[a] Herbert J. Bernstein,[b]* Aaron S. Brewster[c] and Graeme Winter[d]

[a]*DECTRIS Ltd, Täfernweg 1,5405 Baden-Dättwil CH*, [b]*Ronin Institute for Independent Scholarship, c/o NSLS II, Brookhaven National Laboratory, Upton, NY USA*, [c]*Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, USA, and* [d]*Diamond Light Source Ltd, Harwell Science and Innovation Campus, Didcot OX11 0DE, UK*. *E-mail: yayahjb@gmail.com*

(Received 0 XXXXXXX 0000; accepted 0 XXXXXXX 0000)

macromolecular data format; **synchrotron**; **NeXus**; **HDF5**; **NXmx**; **CBF**; **imgCIF**

This is a draft of a Gold Standard paper for IUCrJ as of 8 December 2019. Members of the community who wish to participate in and/or lend their names to this effort are invited to contact

H. J. Bernstein `yayahjb@gmail.com`

Please respond on or before 2 January 2020

Abstract

Macromolecular crystallography (MX) is the dominant means of determining the three-dimensional structures of biological macromolecules. Over the last few decades, most MX data have been collected at synchrotron beamlines using a large number

of different detectors produced by various manufacturers. These data came in their own formats, sometimes proprietary, sometimes open. The associated metadata rarely reached a degree of completion required for data management according to FAIR principles. Efforts to reuse old data by other investigators or even by the original investigators some time later were often frustrated. In the culmination of an effort dating back more than two decades, a large portion of the research community concerned with High Data-Rate Macromolecular Crystallography (HDRMX) has now agreed to an updated specification of data and metadata for diffraction images produced at synchrotron light sources and X-ray free electron lasers (XFELs). This Gold Standard will facilitate processing of datasets at facilities other than the ones at which they were collected and enable data archiving according to FAIR standards, with a particular focus on interoperability and reusability. The agreed standard builds on the NeXuS/HDF5 NXmx application definition and the IUCr CIF imgCIF/CBF dictionary and is compatible with major data processing programs and pipelines.

1. Introduction

"In the 1950's and 1960's, macromolecular crystallographic (MX) data were collected either by precession methods onto film or by single counter diffractometry. ... It was clear that users would benefit from the development of a method that would provide the efficiency of film and the accuracy and automaticity of diffractometery. The "best of both worlds" would thus be a method of electronic detection that combined the advantage of both ... techniques." (Howard, 1996). By the mid 1990's such area detectors had become well-established in MX, but there was lack of agreement on a common format for the data and supporting metadata. In 1995, Andrew Hammersley proposed a "Crystallogrpahic Binary Format", which, after considerable discussion and revision, was adopted by the IUCr in 2005 (Bernstein, 2005)(Bernstein & Hammer-

sley, 2005)(Ellis & Bernstein, 2005). The resulting "imgCIF/CBF" format, metadata and supporting software was adopted by Dectris for the then new PILATUS detector in 2007 (Powell *et al.*, 2007). In subsequent years it became clear that changes would be needed to this format to support higher data rates and institutional policies (Bernstein, 2010). For the Dectris EIGER detectors, CBF was integrated with the Hierarchical Data Format and became the new format NeXus NXmx format for the EIGER detector images. (Donath *et al.*, 2013)(Könnecke *et al.*, 2015)(Bernstein, 2017).

The concepts necessary to sharing data effectively: Findability, Accessability, Interoperability and Reusability (FAIR) have long been recognized, and were formalized recently as "The FAIR Guiding Principles for scientific data management and stewardship" [Wiklinson2016] and are now widely accepted. Both CBF for the PILATUS and NXmx for the EIGER have worked well within the context of data collection at specific beamlines at various facilities, but with the passage of time, variations in the choices of mandatory metadata has created difficulties in processing data collected in a given facility with software in use at other institutions. This has been an ongoing and increasing problem since 2007, especially with respect to interoperability and reusability. This problem has been recognized by a large portion of the research community concerned with High Data-Rate Macromolecular Crystallography (HDRMX). After two decade of effort agreement has been reached on an updated specification of data and metadata for diffraction images to be produced at light sources to facilitate processing of datasets at facilities other than the ones at which they were collected. We call this new specification the "Gold Standard". The agreed specification builds on the NeXUs/HDF5 NXmx application definition and the IUCr CIF imgCIF/CBF dictionary.

2. MX and its history of sharing, openness, standards

There is a natural tension between the desire for a scientist to work on their own data and the value to the field as a whole in seeing as much data as possible shared. Academic macromolecular crystallography has been sharing data on atomic coordinates in standardized formats since the establishment of the Protein Data Bank in 1971 (Bernstein *et al.*, 1977). For macromolecules the PDB coordinate format became the *de facto* standard for MX. Starting in 1990, the small molecule community began a rapid transition to standardized formats for coordinate data using the Crystallographic Information File (CIF) format (Hall *et al.*, 1991). The MX community began a discussion of a macromolecular CIF (mmCIF) for coordinate data in 1993 (Fitzgerald *et al.*, 1993). Diffraction image formats were still fragmented, however. The deposition of structure factors at the PDB was permitted from the beginning. By 1995, one quarter of PDB depositions were made with structure factors in a variety of formats favored by various software packages. By 1996, the fraction of depositions with structure factors had risen to more than half and use of an mmCIF-based standard format for structure factors was agreed (Jiang *et al.*, 1999). As noted in section 1, at the same time the MX community began serious consideration of imgCIF/CBF as a standardized, open format for diffraction images.

2.1. A history of incomplete and incompatible metadata

The process of adoption of of a standardized open diffraction image has been slow. One of the most difficult to surmount potential barriers to adoption of a common format has been lack of agreement about what metadata should always be incorporated with diffraction image data. For some experiments and processing programs only the image itself is needed. All other data and metadata, such as wavelength, detector distance, rotation angles, etc. are provided separately in ".INP" or ".site" files. When

the Pilatus CBF image format was adopted in 2007 it was specified with complete metadata, but shortly after that the so-called "miniCBF" format with much more limited metadata was adopted and has been widely used (Dectris, 2013). Because the limited list of metadata in one miniCBF collected to the standards of one facility may not be sufficient to meet the processing demands of software at other facilities, a large number of undocumented variants of the miniCBF format with idiosyncratic and inconsistent metadata have cropped up, necessitating unanticipated searches through laboratory notebooks and other records to resolve ambiguities, as well as site-specific patched to software. When data collections took days to weeks of beam-time and computer time, this was viewed as a minor issue at many facilities with the occasional nuisance of searching for missing metadata being a reasonable cost to pay for the convenience of a short, simple list of required metadata.

2.2. Transformation of MX with HPC detectors with high data rates

In 2007 the first Pilatus detectors strained then-available computers and networks with ten six-megapixel frames per second. Now Eiger detectors are capable of 133 eighteen-megapixel frames per second, and the latest Eiger 2 XE `https://www.dectris.com/products/eiger/eiger2-x-xe-for-synchrotron/BR_EIGER2_XE_Jul2019_r0.pdf` can do 400-550 frames per second. With new, smaller more intense beamlines, data collections now run two orders of magnitude faster than just a few years ago, with every prospect of higher data rates to come. As many collections as possible need to be fully automated. The time that used to be lost to dealing with incomplete or inconsistent metadata is no longer acceptable.

2.3. Hardware, Software, Automation and the need for standards

While internal computer component interconnects, such as the AMD infinity fabric `https://wccftech.com/amds-infinity-fabric-detailed/` are being engineered to reach speeds of .5 to 4 Tbps, and true terabit Ethernet is expected to become a reality by 2025 (`https://en.wikipedia.org/wiki/Terabit_Ethernet`) at present the best we can expect at present at reasonable cost are PCI Express 4 (.25 Tbps) and PCI Express 5 (.5 Tbps) internal connects and .1 Tbps, .2 Tbps and .4 Tbps LAN speeds. As a consequence the major bottleneck in diffraction image processing is the motion of data. Any unnecessary transfers or conversions of image data need to be avoided. In addition, most of the software in current use was designed in a context of processors supporting very little parallelism, even though the increasing demand for automation in response to higher detector speeds and more intense beams can only be satisfied by higher levels of parallelism, but the necessary algorithmic changes are challenging to address. We are in the peculiar position that the easiest step to take to meet the need for higher performance is to adopt uniform standards for data and metadata so that as few conversions and data motions are possible are needed.

2.4. Data archiving (FAIR)

While the immediate benefit for uniform MX standards is in achieving the best performance, uniform MX data and metadata standards also make easier to prepare datasets for archiving (Helliwell, 2019). This then facilitates reuse of the raw data, both for better processing with future improved methods, and in the use of crystallographic structures for molecular replacement in other crystallographic structures or as higher resolution components of cryoEM images of large molecular machines.

3. History of HDRMX meetings and development of the idea of a Gold Standard

Starting in 2016 beamline scientists, controls people and others with an interest in High Data-Rate Macromolecular Crystallography have been meeting on an irregular schedule to explore way to improve the processing of crystallographic data from the newest generations of detectors (see `http://hdrmx.medsbio.org`). The effort that led to the "Gold Standard" began in the HDRMX meeting at ACA 2018, in Toronto, CA 22 July 2018, continued with further discussion at the HDRMX Satellite to AsCA 2018/Crystal 32 in Auckland, NZ 6 - 7 December 2018, at the HDRMX meeting at ACA 2019, Covington, KY 21 July 2019, the HDRMX meeting at ECM 32, Vienna, Austria at 7:30 pm on 20 August 2019, and achieved final agreement on the Gold Standard at the HDRMX meeting at Diamond Light Source, Chilton, UK 6-7 November 2019.

4. Description of Gold Standard, compliance with software

Whether we are dealing with Pilatus CBF files or Eiger NeXus/HDF5 files, the information in a "Gold Standard" dataset is the same: one or more diffraction image data arrays of pixels along with enough metadata to allow software to determine exactly where in the laboratory coordinate system each pixel was located and when the intensity recorded in that pixel was recorded, so that the software can locate spots, index them and integrate them. In the past some of the metadata needed for this process might have been recorded in the same set of files as the image data arrays and some of the necessary metadata might be recorded elsewhere, e.g. in a laboratory notebook, or some separate electronic laboratory notebook. In a Gold Standard dataset, all the necessary data and metadata for processing is recorded in the dataset, so that the dataset can be moved freely to other filesystems in other facilities and still be pro-

cessed without the need to return to the original facility to find metadata that had been left behind. Though the dataset may, indeed, normally will consist of multiple files, those files should be packaged together in some appropriate container, e.g. a single folder in the collecting facilty's file system or under a single DOI in a dataset repository.

The specification of what metadata need be retained with the data depends on the experiment being performed and the software that will be used for processing, *i.e.* the "use case". The "Gold Standard" being discussed here is intended to be adequate for single-axis rotation experiments at synchrotrons and stills collected at XFELs and to support DIALS (Waterman *et al.*, 2013)(Winter *et al.*, 2018), XDS (Kabsch, 2010*b*)(Kabsch, 2010*a*), Mosflm(Battye *et al.*, 2011), HKL2000 (Otwinowski & Minor, 1997) and autoProc (Vonrhein *et al.*, 2011). The more complex the design of the experiment and the more varied the non-default choices permitted by the software, the more different metadata may be required to ensure correct processing at a wide range of facilities. The following is the minimum mandatory set of metadata agreed upon.

4.1. Identifying the Provenance of the Data

While each dataset should "stand on it own legs" and contain all the data and metadata necessary for processing, it also should clearly identify where and when it was collected, by specifying the beamline and facility at which it was collected and the times of collection. In the NXmx Gold Standard, the full name of the beamline is carried in the `/(entry):NXentry/(instrument):NXinstrument/name` field, the the name of the facility is carried in the `/(entry):NXentry/(source):NXsource/name` field. The commonly used acronyms or abbreviations of each of the names in those `name` fields are carried in the associated `@short_name` attributes. The full and precise UTC ISO 8601

(Wolf & Wicksteed, 1998) time/date of the first data point collected is carried in the `/(entry):NXentry/start_time` field and an estimate of the likely time of collection of the last data point is carried in the `/(entry):NXentry/end_time_estimated` field. If/when the data collection is completed, the full and precise UTC ISO 8601 time/date of the last data point collected is carried in the `/(entry):NXentry/end_time` field. The time zone of the beamline is carried in the `/(entry):NXentry/(instrument):NXinstrument/end_time` field so local times may be recovered.

4.2. Identifying Where Components are Positioned and How they are Oriented

One of the most important sets of metadata used for processing is information on where components in the experimental setup are positioned and oriented relative to one another. We would like to know how the sample is positioned and oriented relative to the incident beam. We would like to know where the detector is positioned and oriented relative to the sample. We would like to know where in or on the detector the incident beam would have hit the detector, and where the various sensor modules of the detector are positioned relative to one another. Essentially we need a blueprint of the experimental setup. The metadata used for this purpose both in CBF and in NeXus/HDF5 describes fixed or variable positioning axes in terms of directional vectors in nested lists with optional offset vectors between pairs of axes. For an experiment with both a detector and a sample goniometer, we need to provide the nested chain of axes that determine to position and orientation of the detector and the nested chain of axes that determine to position and orientation of the sample. In each case we do this backwards, starting with a specification in the description of the detector of a `depends_on` field specifying the axis that actually supports the detector and a specification in the description of the sample of a `depends_on` field specifying the axis that actually supports the sample. For each axis that is supported by another axis,

we describe that axis next, until be reach a fixed point in the beamline, denoted by a ".". For both CBF and NeXus, the origin of the coordinate system used is intended to be in the sample. When there is a rotation axis for the sample, the origin is at the intersection of the beam and that axis. If there is no sample rotation axis, the midpoint of the line segment marking the intersection of the beam with the sample is usually used. The axes of the NeXus/HDF5 coordinate system are described in Fig. 1 and the axes of the CBF coordinate system are described in Fig. 2.

All axis chain definitions and axis settings necessary to process the data should be clearly and explicitly described. In a NeXus/HDF5 NXmx file these descriptions begin with the `depends_on` field and `NXtransformations` group in each `NXdetector` group and in each `NXsample` group. In addition the axis of the beam direction and of the downward direction of gravity will be specified, because they are needed in the McStas coordinate system used in NeXus.

The axes pointed to from each `depends_on` field should be placed in appropriate NXtransformations groups. Each axis has a dimensionless unit vector and an optional offset vector specifying the direction cosines of the axis and the offset from the previous axis in the chain to the base of the new axis.

The NeXus/HDF5 files specify axes in the NeXus McStas coordinate system. It is important to note that imgCIF/CBF uses a different coordinate system. The standard coordinate frame in NeXus is the McStas coordinate frame (Lefmann & Nielsen, 1999), in which the Z axis points in the direction of the incident beam, the X axis is orthogonal to the Z axis in the horizontal plane and pointing left as seen from the source and the Y axis points upwards. The origin is in the sample.

The standard coordinate frame in imgCIF/CBF aligns the X axis to the principal goniometer axis, chooses the Z axis to point from the sample into the beam. If the beam is not orthogonal to the X axis, the Z axis is the component orthogonal to

the X axis the of "-Beam" vector. The "-Beam" vector is the negative of the "Beam" vector, *i.e.* a vector which points towards the source. The Y axis is chosen to complete a right-handed axis system. The origin is in the sample.

Appendix A
Supplement: GoldStandard NeXus NXmx application definition

This is a snapshot of the HDRMX NXmx application definition as it is being proposed to the NeXus International Advisory Committee (NIAC) for adoption by NIAC. The latest version prior prior to formal adoption is available from `http://github.com/HDRMX/definitions`.

Status:

application definition, extends NXobject

Description:

functional application definition for macromolecular crystallography

Symbols:

These symbols will be used below to coordinate datasets with the same shape. Most MX x-ray detectors will produce two-dimensional images. Some will produce three-dimensional images, using one of the indices to select a detector element.

dataRank: rank of the `data` field

np: number of scan points

i: number of detector pixels in the slowest direction

j: number of detector pixels in the second slowest direction

k: number of detector pixels in the third slowest direction

Groups cited: NXattenuator, NXbeam, NXcollection, NXdata, NXdetector_group, NXdetector_module, NXdetector, NXentry, NXgeometry, NXinstrument, NXnote, NXsample, NXsource, NXtransformations

Structure:

(entry): (required) NXentry

title: (optional) NX_CHAR

start_time: (recommended) NX_DATE_TIME

ISO 8601 time/date of the first data point collected in UTC, using the Z suffix to avoid confusion with local time. Note that the time zone of the beamline should be provided in NXentry/NXinstrument/time_zone.

end_time: (optional) NX_DATE_TIME

ISO 8601 time/date of the last data point collected in UTC, using the Z suffix to avoid confusion with local time. Note that the time zone of the beamline should be provided in NXentry/NXinstrument/time_zone. This field should only be filled when the value is accurately observed.

end_time_estimated: (optional) NX_DATE_TIME

ISO 8601 time/date of the last data point collected in UTC, using the Z suffix to avoid confusion with local time. Note that the time zone of the beamline should be provided in NXentry/NXinstrument/time_zone. This field may be filled with a value estimated before an observed value is avilable.

definition: (required) NX_CHAR

NeXus NXDL schema to which this file conforms

Obligatory value: `NXmx`

(data): (required) NXdata

data[np, i, j, k]: (recommended) NX_NUMBER

For a dimension-2 detector, the rank of the data array will be 3. For a dimension-3 detector, the rank of the data array will be 4. This allows for the introduction of the frame number as the first index.

(sample): (required) NXsample

name: (required) NX_CHAR

Descriptive name of sample

depends_on: (required) NX_CHAR

This is a requirement to describe for any scan experiment.

The axis on which the sample position depends may be stored anywhere, but is normally stored in the NXtransformations group within the NXsample group.

If there is no goniometer, e.g. with a jet, depends_on should be set to .

temperature: (optional) NX_CHAR {units=NX_TEMPERATURE}

(transformations): (required) NXtransformations

This is the recommended location for sample goniometer and other related axes.

This is a requirement to describe for any scan experiment. The reason it is optional is mainly to accommodate XFEL single shot exposures.

Use of the depends_on field and the NXtransformations group is strongly recommended. As noted above this should be an absolute requirement to have for any scan experiment.

The reason it is optional is mainly to accommodate XFEL single shot exposures.

(instrument): (required) NXinstrument

name: (required) NX_CHAR

Name of instrument

@short_name: (required) NX_CHAR

short name for instrument, perhaps the acronym

time_zone: (recommended) NX_DATE_TIME

ISO 8601 time_zone offset from UTC

(attenuator): (optional) NXattenuator

attenuator_transmission: (optional) NX_NUMBER {units=NX_UNITLESS}

(detector_group): (recommended) NXdetector_group

Optional logical grouping of detector elements.

Each detector element is represented as an NXdetector group with its own detector data array. Each detector data array may be further decomposed into array sections by use of NXdetector_module groups. The names are given in the group names field.

The groups are defined hierarchically, with names given in the group_names field, unique identifiing indices given in the field group_index, and the level

in the hierarchy given in the group_parent field. For example if an x-ray detector, DET, consists of four elements in a rectangular array:

```
DTL    DTR
DLL    DLR
```

We could have:

```
group_names:
  ["DET", "DTL", "DTR", "DLL", "DLR"]
group_index: [1, 2, 3, 4, 5]
group_parent:  [-1, 1, 1, 1, 1]
```

group_names: (required) NX_CHAR

An array of the names of the detector elements or hierarchical groupings of detector elements.

Specified in the base classes as comma separated list of names, but new code should use an array of names as quoted strings.

group_index[i]: (required) NX_INT

An array of unique indices for detector elements or groupings of detector elements.

Each element is a unique ID for the corresponding group named in the field group_names. The IDs are positive integers starting with 1.

group_parent[group_index]: (required) NX_INT

An array of the hierarchical levels of the parents of detector elements or groupings of detector elements.

A top-level element or grouping has parent level -1

(detector): (required) NXdetector

Normally the detector group will have the name `detector`. However, in the case of multiple detector elements, each element needs a uniquely named NXdetector group.

depends_on: (required) NX_CHAR

NeXus path to the detector positioner axis that most directly supports the detector.

data[np, i, j, k]: (recommended) NX_NUMBER

For a dimension-2 detector, the rank of the data array will be 3. For a dimension-3 detector, the rank of the data array will be 4. This allows for the introduction of the frame number as the first index.

description: (recommended) NX_CHAR

name/manufacturer/model/etc. information

time_per_channel: (optional) NX_CHAR {units=NX_TIME}

todo: define more clearly

distance: (recommended) NX_FLOAT {units=NX_LENGTH}

Distance from the sample to the beam center. Normally this value is for guidance only, the proper geometry can be found following the depends_on axis chain, But in appropriate cases where the dectector distance to the sample is observable independent of the axis chain, that may take precedence over the axis chain calculation.

distance_derived: (recommended) NX_BOOLEAN {units=NX_LENGTH}

Boolean to indicate if the distance is a derived, rather than a primary observation. If distance_derived true or is not specified, the distance is assumed to be derived from delector axis specifications.

dead_time: (optional) NX_FLOAT {units=NX_TIME}

Detector dead time

count_time: (recommended) NX_NUMBER {units=NX_TIME}

Elapsed actual counting time

beam_center_derived: (optional) NX_BOOLEAN {units=NX_LENGTH}

Boolean to indicate if the distance is a derived, rather than a primary observation. If true or not provided, that value of beam_center_derived is assumed to be true

beam_center_x: (recommended) NX_FLOAT {units=NX_LENGTH}

This is the x position where the direct beam would hit the detector. This is a length and can be outside of the actual detector. The length can be in physical units or pixels as documented by the units attribute. Normally, this should be derived from the axis chain, but the direct specification may take precedence if it is not a derived quantity.

beam_center_y: (recommended) NX_FLOAT {units=NX_LENGTH}

This is the y position where the direct beam would hit the detector. This is a length and can be outside of the actual detector. The length can be in physical units or pixels as documented by the units attribute. Normally, this should be derived from the axis chain, but the direct specification may take precedence if it is not a derived quantity.

angular_calibration_applied: (optional) NX_BOOLEAN

True when the angular calibration has been applied in the electronics, false otherwise.

angular_calibration[i, j, k]: (optional) NX_FLOAT

Angular calibration data.

flatfield_applied: (optional) NX_BOOLEAN

True when the flat field correction has been applied in the electronics, false otherwise.

flatfield[i, j, k]: (optional) NX_FLOAT

Flat field correction data. If provided, it is recommended that is be compressed

flatfield_error[i, j, k]: (optional) NX_FLOAT

Errors of the flat field correction data. If provided, it is recommended that it be compressed

pixel_mask_applied: (optional) NX_BOOLEAN

True when the pixel mask correction has been applied in the electronics, false otherwise.

pixel_mask[i, j]: (recommended) NX_INT

The 32-bit pixel mask for the detector. Can be either one mask for the whole dataset (i.e. an array with indices i, j) or each frame can have

its own mask (in which case it would be an array with indices np, i, j). Contains a bit field for each pixel to signal dead, blind or high or otherwise unwanted or undesirable pixels. They have the following meaning:

- bit 0: gap (pixel with no sensor)
- bit 1: dead
- bit 2: under responding
- bit 3: over responding
- bit 4: noisy
- bit 5: -undefined-
- bit 6: pixel is part of a cluster of problematic pixels (bit set in addition to others)
- bit 7: -undefined-
- bit 8: user defined mask (e.g. around beamstop)
- bits 9-30: -undefined-
- bit 31: virtual pixel (corner pixel with interpolated value)

Normal data analysis software would not take pixels into account when a bit in (mask & 0x0000FFFF) is set. Tag bit in the upper two bytes would indicate special pixel properties that normally would not be a sole reason to reject the intensity value (unless lower bits are set.

If the full bit depths is not required, providing a mask with fewer bits is permissible.

If needed, additional pixel masks can be specified by including additional entries named pixel_mask_N, where N is an integer. For example, a general bad pixel mask could be specified in pixel_mask that indicates

noisy and dead pixels, and an additional pixel mask from experiment-specific shadowing could be specified in pixel_mask_2. The cumulative mask is the bitwise OR of pixel_mask and any pixel_mask_N entries.

If provided, it is recommended that it be compressed

countrate_correction_applied: (optional) NX_BOOLEAN

True when a count-rate correction has already been applied in the data recorded here, false otherwise.

bit_depth_readout: (recommended) NX_INT

How many bits the electronics record per pixel.

detector_readout_time: (optional) NX_FLOAT {units=NX_TIME}

Time it takes to read the detector (typically milliseconds). This is important to know for time resolved experiments.

frame_time: (optional) NX_FLOAT {units=NX_TIME}

This is time for each frame. This is exposure_time + readout time.

gain_setting: (optional) NX_CHAR

The gain setting of the detector. This influences background.

saturation_value: (optional) NX_INT

The value at which the detector goes into saturation. Data above this value is known to be invalid.

sensor_material: (required) NX_CHAR

At times, radiation is not directly sensed by the detector. Rather, the detector might sense the output from some converter like a scintillator. This is the name of this converter material.

sensor_thickness: (required) NX_FLOAT {units=NX_LENGTH}

At times, radiation is not directly sensed by the detector. Rather, the detector might sense the output from some converter like a scintillator. This is the thickness of this converter material.

threshold_energy: (optional) NX_FLOAT {units=NX_ENERGY}

Single photon counter detectors can be adjusted for a certain energy range in which they work optimally. This is the energy setting for this.

type: (optional) NX_CHAR

Description of type such as scintillator, ccd, pixel, image plate, CMOS, ...

(transformations): (required) NXtransformations

Location for axes (transformations) to do with the detector

(collection): (optional) NXcollection

Suggested container for detailed non-standard detector information like corrections applied automatically or performance settings.

(detector_module): (required) NXdetector_module

Many detectors consist of multiple smaller modules that are operated in sync and store their data in a common dataset. To allow consistent

parsing of the experimental geometry, this application definiton requires all detectors to define a detector module, even if there is only one.

This group specifies the hyperslab of data in the data array associated with the detector that contains the data for this module. If the module is associated with a full data array, rather than with a hyperslab within a larger array, then a single module should be defined, spanning the entire array.

data_origin: (required) NX_INT

A dimension-2 or dimension-3 field which gives the indices of the origin of the hyperslab of data for this module in the main area detector image in the parent NXdetector module.

The data_origin is 0-based.

The frame number dimension (np) is omitted. Thus the data_origin field for a dimension-2 dataset with indices (np, i, j) will be an array with indices (i, j), and for a dimension-3 dataset with indices (np, i, j, k) will be an array with indices (i, j, k).

The order of indices (i, j or i, j, k) is slow to fast.

data_size: (required) NX_INT

Two or three values for the size of the module in pixels in each direction. Dimensionality and order of indices is the same as for data_origin.

data_stride: (optional) NX_INT

Two or three values for the stride of the module in pixels in each direction. By default the stride is [1,1] or [1,1,1], and this is the most likely case. This optional field is included for completeness.

module_offset: (optional) NX_NUMBER {units=NX_LENGTH}

Offset of the module in regards to the origin of the detector in an arbitrary direction.

@transformation_type: (required) NX_CHAR

Obligatory value: `translation`

@vector: (required) NX_CHAR

@offset: (required) NX_CHAR

@depends_on: (required) NX_CHAR

fast_pixel_direction: (required) NX_NUMBER {units=NX_LENGTH}

Values along the direction of fastest varying pixel direction. The direction itself is given through the vector attribute

@transformation_type: (required) NX_CHAR

Obligatory value: `translation`

@vector: (required) NX_CHAR

@offset: (required) NX_CHAR

@depends_on: (required) NX_CHAR

slow_pixel_direction: (required) NX_NUMBER {units=NX_LENGTH}

Values along the direction of slow varying pixel direction. The direction itself is given through the vector attribute

@transformation_type: (required) NX_CHAR

Obligatory value: `translation`

@vector: (required) NX_CHAR

@offset: (required) NX_CHAR

@depends_on: (required) NX_CHAR

(beam): (required) NXbeam

incident_wavelength: (required) NX_FLOAT {units=NX_WAVELENGTH}

In the case of a monchromatic beam this is the scalar wavelength.

In the case of a polychromatic beam this is an array of the wavelengths with the relative weights in incident_wavelength_weight.

incident_wavelength_weight: (optional) NX_FLOAT

In the case of a polychromatic beam this is an array of the relative weights of the corresponding wavelengths in incident_wavelength.

incident_wavelength_spread: (optional) NX_FLOAT {units=NX_WAVELENGTH}

The wavelength spread FWHM for the corresponding wavelength(s) in incident_wavelength.

flux: (optional) NX_FLOAT {units=NX_FLUX}

flux incident on beam plane area in photons per second per unit area

total_flux: (required) NX_FLOAT {units=NX_FREQUENCY}

flux incident on beam plane in photons per second

incident_beam_size[2]: (recommended) NX_FLOAT {units=NX_LENGTH}

Two-element array of FWHM (if Gaussian or Airy function) or diameters (if top hat) or widths (if rectangular) of beam in the order x, y

profile: (recommended) NX_CHAR

The beam profile, Gaussian, Airy function, top-hat or rectangular. The profile is given in the plane of incidence of the beam on the sample.

Any of these values: `Gaussian` — `Airy` — `top-hat` — `rectangular`

incident_polarisation_stokes[np, 4]: (recommended) NX_CHAR

incident_wavelength_spectrum: (optional) NXdata

(source): (required) NXsource

The neutron or x-ray storage ring/facility.

distance: (optional) NX_FLOAT {units=NX_LENGTH}

Effective distance from sample Distance as seen by radiation from sample. This number should be negative to signify that it is upstream of the sample.

name: (required) NX_CHAR

Name of source

@short_name: (optional) NX_CHAR

short name for source, perhaps the acronym

type: (optional) NX_CHAR

type of radiation source (pick one from the enumerated list and spell exactly)

Any of these values:

- `Spallation Neutron Source`
- `Pulsed Reactor Neutron Source`
- `Reactor Neutron Source`

- `Synchrotron X-ray Source`
- `Pulsed Muon Source`
- `Rotating Anode X-ray`
- `Fixed Tube X-ray`
- `UV Laser`
- `Free-Electron Laser`
- `Optical Laser`
- `Ion Source`
- `UV Plasma Source`

probe: (optional) NX_CHAR

type of radiation probe (pick one from the enumerated list and spell exactly)

Any of these values:

- `neutron`
- `x-ray`
- `muon`
- `electron`
- `ultraviolet`
- `visible light`
- `positron`
- `proton`

power: (optional) NX_FLOAT {units=NX_POWER}

Source power

emittance_x: (optional) NX_FLOAT {units=NX_EMITTANCE}

Source emittance (nm-rad) in X (horizontal) direction.

emittance_y: (optional) NX_FLOAT {units=NX_EMITTANCE}

Source emittance (nm-rad) in Y (horizontal) direction.

sigma_x: (optional) NX_FLOAT {units=NX_LENGTH}

particle beam size in x

sigma_y: (optional) NX_FLOAT {units=NX_LENGTH}

particle beam size in y

flux: (optional) NX_FLOAT {units=NX_FLUX}

Source intensity/area (example: s-1 cm-2)

energy: (optional) NX_FLOAT {units=NX_ENERGY}

Source energy. For storage rings, this would be the particle beam energy. For X-ray tubes, this would be the excitation voltage.

current: (optional) NX_FLOAT {units=NX_CURRENT}

Accelerator, X-ray tube, or storage ring current

voltage: (optional) NX_FLOAT {units=NX_VOLTAGE}

Accelerator voltage

frequency: (optional) NX_FLOAT {units=NX_FREQUENCY}

Frequency of pulsed source

period: (optional) NX_FLOAT {units=NX_PERIOD}

Period of pulsed source

target_material: (optional) NX_CHAR

Pulsed source target material

Any of these values:

- `Ta`
- `W`
- `depleted_U`
- `enriched_U`
- `Hg`
- `Pb`
- `C`

number_of_bunches: (optional) NX_INT

For storage rings, the number of bunches in use.

bunch_length: (optional) NX_FLOAT {units=NX_TIME}

For storage rings, temporal length of the bunch

bunch_distance: (optional) NX_FLOAT {units=NX_TIME}

For storage rings, time between bunches

pulse_width: (optional) NX_FLOAT {units=NX_TIME}

temporal width of source pulse

mode: (optional) NX_CHAR

source operating mode

Any of these values:

- `Single Bunch`: for storage rings
- `Multi Bunch`: for storage rings

top_up: (optional) NX_BOOLEAN

Is the synchrotron operating in top_up mode?

last_fill: (optional) NX_NUMBER {units=NX_CURRENT}

For storage rings, the current at the end of the most recent injection.

@time: (required) NX_DATE_TIME

date and time of the most recent injection.

notes: (optional) NXnote

any source/facility related messages/events that occurred during the experiment

bunch_pattern: (optional) NXdata

For storage rings, description of the bunch pattern. This is useful to describe irregular bunch patterns.

title: (required) NX_CHAR

name of the bunch pattern

pulse_shape: (optional) NXdata

source pulse shape

geometry: (optional) NXgeometry

Engineering location of source

distribution: (optional) NXdata

The wavelength or energy distribution of the source

Acknowledgements

The work reported is the result of the efforts of many people for the past several decades. Particular credit is due to Andrew P. Hammersley who made the initial CBF proposal in 1995, and to the patient work of all the participants in the many HDRMX workshops.

References

Battye, T. G. G., Kontogiannis, L., Johnson, O., Powell, H. R. & Leslie, A. G. W. (2011). *Acta Cryst.* D**67**(4), 271 – 281.

Bernstein, F. C., Koetzle, T. F., Williams, G. J. B., Meyer, Jr., E. F., Brice, M. D., Rodgers, J. R., Kennard, O., Shimanouchi, T. & Tasumi, M. (1977). *J. Mol. Biol.* **112**, 535 – 542.

Bernstein, H. J. (2005). *Definition and Exchange of Crystallographic Data, International Tables For Crystallography*, chap. Classification and Use of Image Data, pp. 199 – 205. International Union of Crystallography, Springer, Dordrecht, NL.

Bernstein, H. J. (2010). HDF5 as hyperspectral data analysis format Workshop, ESRF, Grenoble, FR, 11-13 January 2010.

Bernstein, H. J. (2017). *Foundations of Crystallography*, **73**, a189.

Bernstein, H. J. & Hammersley, A. P. (2005). In *International Tables For Crystallography*, edited by S. R. Hall & B. McMahon, vol. G: Definition and Exchange of Crystallographic Data, chap. 2.3, pp. 37 – 43. International Union of Crystallography, Springer, Dordrecht, NL.

Dectris, (2013). *PILATUS CBF Header Specification*. Dectris Ltd., DECTRIS Ltd., Neuenhoferstrasse 107, 5400 Baden, Switzerland, 1st ed. `https://www.dectris.com/support/downloads/header-docs/cbf/Pilatus_CBF_Header_Specification-4.pdf`.

Donath, T., Rissi, M. & Billich, H. (2013). *Synchrotron Radiation News*, **26**(5), 34 – 35.

Ellis, P. J. & Bernstein, H. J. (2005). *Definition and Exchange of Crystallographic Data, International Tables For Crystallography*, chap. CBFlib: an ANSI C library for manipulating image data, pp. 544 – 556. International Union of Crystallography, Springer, Dordrecht, NL.

Fitzgerald, P., Berman, H., Bourne, P., Watenpaugh, K. & Westbrook, J. (1993). In *American Crystallographic Association Annual Meeting, Albuquerque, NM USA*.

Hall, S. R., Allen, F. H. & Brown, I. D. (1991). *Acta Crystallographica Section A: Foundations of Crystallography*, **47**(6), 655 – 685.

Helliwell, J. R. (2019). *Structural Dynamics*, **6**(5), 054306.

Howard, A. J. (1996). *Crystallographic computing 7: Proceedings from the macromolecular crystallographic computing school*.

Jiang, J., Abola, E. & Sussman, J. L. (1999). *Acta Cryst.* D**55**(1), 4.

Kabsch, W. (2010*a*). *Acta Crystallographica Section D: Biological Crystallography*, **66**(2), 133–144.

Kabsch, W. (2010*b*). *Acta Cryst.* D**66**(2), 125 – 132.

Könnecke, M., Akeroyd, F. A., Bernstein, H. J., Brewster, A. S., Campbell, S. I., Clausen, B., Cottrell, S., Hoffmann, J. U., Jemian, P. R., Männicke, D. *et al.* (2015). *Journal of applied crystallography*, **48**(1), 301 – 305.

Lefmann, K. & Nielsen, K. (1999). *Neutron news*, **10**(3), 20 – 23.

Otwinowski, Z. & Minor, W. (1997). In *Methods in Enzymology*, vol. 276, pp. 307 – 326. Elsevier.

Powell, H., Leslie, A. & Battye, G. (2007). *CCP4 Newsletter on Protein Crystallography*, (46).

Vonrhein, C., Flensburg, C., Keller, P., Sharff, A., Smart, O., Paciorek, W., Womack, T. & Bricogne, G. (2011). *Acta Cryst.* D**67**(4), 293 – 302.

Waterman, D. G., Winter, G., Parkhurst, J. M., Fuentes-Montero, L., Hattne, J., Brewster, A., Sauter, N. K., Evans, G. & Rosenstrom, P. (2013). *CCP4 Newslett. Protein Crystallogr*, **49**, 13 – 15.

Winter, G., Waterman, D. G., Parkhurst, J. M., Brewster, A. S., Gildea, R. J., Gerstel, M., Fuentes-Montero, L., Vollmar, M., Michels-Clark, T., Young, I. D. *et al.* (2018). *Acta Cryst.* D**74**(2), 85 – 97.

Wolf, M. & Wicksteed, C. (1998). *W3C NOTE NOTE-datetime-19980827, August*, p. 26.

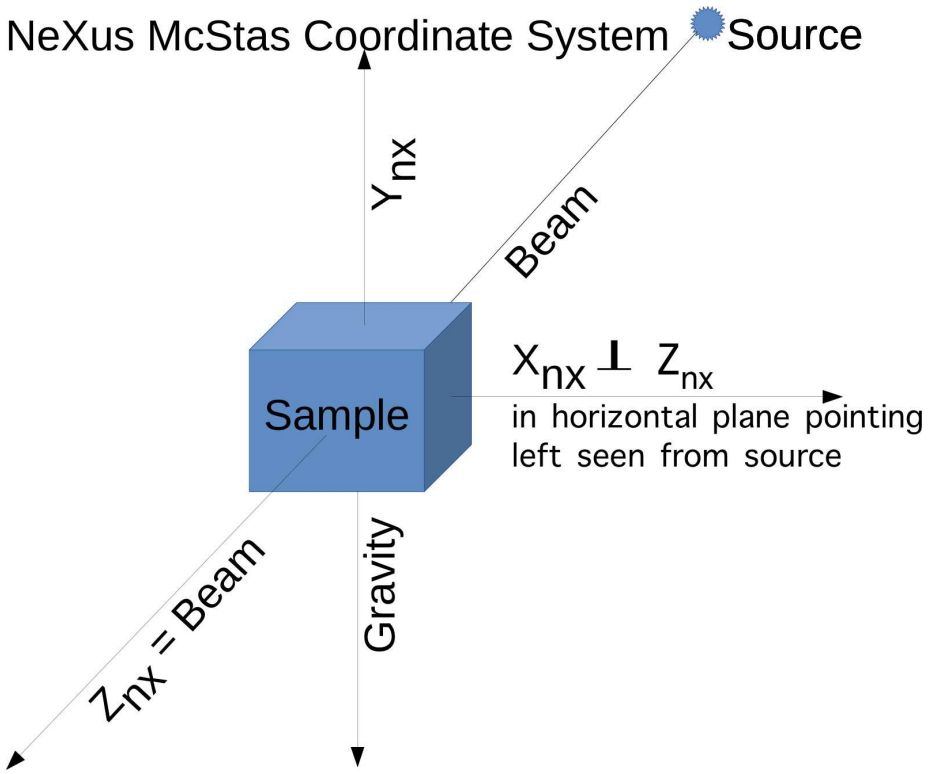

Fig. 1. The NeXus/HDF5 files specify axes in the NeXus McStas coordinate system. The standard coordinate frame in NeXus is the McStas coordinate frame, in which the Z axis points in the direction of the incident beam, the X axis is orthogonal to the Z axis in the horizontal plane and pointing left as seen from the source and the Y axis points upwards. The origin is in the sample.

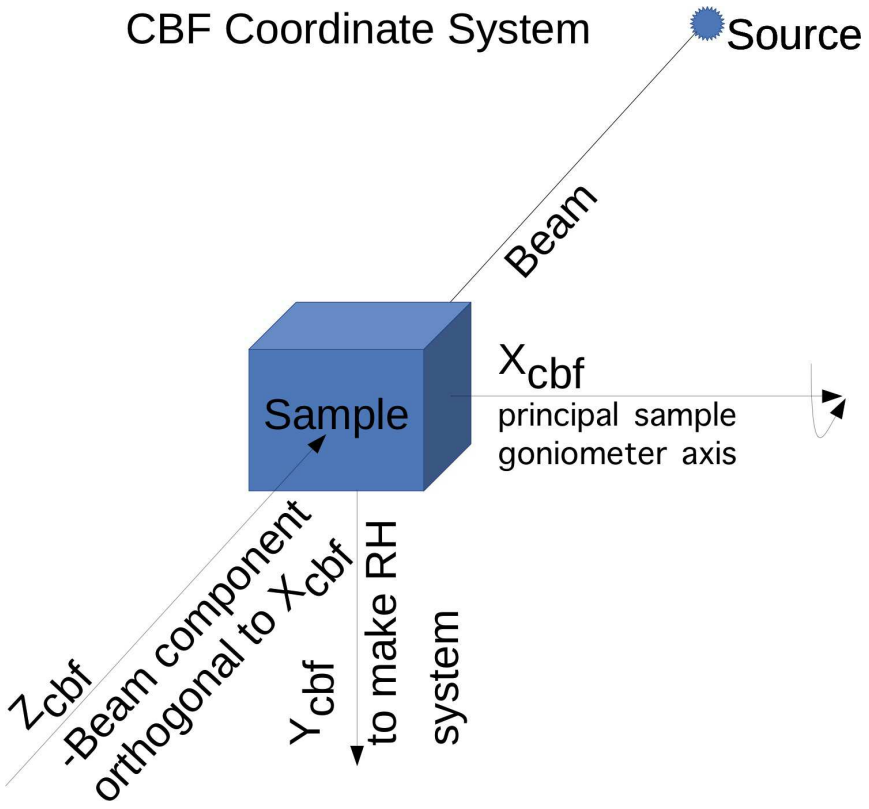

Fig. 2. The standard coordinate frame in imgCIF/CBF aligns the X axis to the principal goniometer axis, chooses the Z axis to point from the sample into the beam. If the beam is not orthogonal to the X axis, the Z axis is the component orthogonal to the X axis the of "-Beam" vector. The "-Beam" vector is the negative of the "Beam" vector, *i.e.* a vector which points towards the source. The Y axis is chosen to complete a right-handed axis system.

Synopsis

A large portion of the research community concerned with High Data-Rate Macromolecular Crystallography (HDRMX) has agreed to an updated specification of data and metadata for diffraction images to be produced at light sources to facilitate processing of datasets at facilities other than the ones at which they were collected and to enable data archiving according to FAIR principles.