ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-THMBCM034

ULTRA-HIGH THROUGHPUT MACROMOLECULAR CRYSTALLOGRAPHY DATA COLLECTION USING THE BLUESKY FRAMEWORK

D. Perl†, N. Frisina, D. Oram, N. Paterson, Diamond Light Source, Didcot, UK

Abstract

At Diamond Light Source (DLS), several Macromolecular Crystallography (MX) beamlines focus on, or include in their activities, completely automated data collection. This is used primarily for high throughput collection on samples with known or partially known structures, for example, screening a protein for drug or drug fragment interactions. The automated data collection routines are currently built on legacy experiment orchestration software which includes a lot of redundancy originally implemented for safety when human users are controlling the beamline, but which is inefficient when the beamline hardware occupies a smaller number of known states. DLS is building its next generation, service-based, Data Acquisition (DAQ) platform, Athena, using NSLS-II's *Bluesky* experiment orchestration library. The *Bluesky* library facilitates optimising the orchestration of experiment control by simplifying the work necessary to parallelise and reorganise the steps of an experimental procedure. The MX data acquisition team at Diamond is using the Athena platform to increase the possible rate of automated MX data collection both for immediate use and in preparation to take advantage of the upgraded Diamond-II synchrotron, due in several years. This project, named *Hyperion*, will include sample orientation and centring, fluorescence scanning, optical monitoring, collection strategy determination, and rotation data collection at multiple positions on a single sample pin.

INTRODUCTION AND MOTIVATION

Motivation

In order to take advantage of the increased beam intensity which will be available following the Diamond-II upgrade [1] and facilitate research which depends on high-throughput techniques we are developing a new unattended data collection (UDC) application. This application is named *Hyperion*, and is built on the *Bluesky* [2] library developed at Brookhaven National Laboratory. *Hyperion* is envisaged to allow the rapid collection of thousands of single-crystal X-ray diffraction datasets per day. This quantity of data is desired in order to screen thousands of candidate molecules against proteins to identify candidates for drugs or to probe structure-function relationships of those proteins.

In addition to the above, we aim for *Hyperion* to help support DLS users in the dark period during the machine upgrade by allowing us to process their samples as usual with much more limited beamtime at another facility. In the nearer future, indeed, as is already being implemented, *Hyperion* improves the existing unattended data collection (UDC) capability at Diamond beamline I03.

Current Data Collection Software

At Diamond, for the last 20 years since the beginning of our operation, we have used the Generic Data Acquisition (GDA) [3] platform. The generic nature of this platform has allowed us to reuse software across beamlines. Data acquisition, in the context of synchrotron beamlines, refers in general to the manipulation of the position of scientific samples (whether biological or material) in an X-ray or other photon beam, and the detection of transmitted, diffracted, reflected or radiated photons from the sample. This includes techniques such as X-ray crystallography, scattering techniques, imaging techniques, and absorption, fluorescence, infrared, or Raman spectroscopy.

GDA has a monolithic architecture and is, in general, not built using contemporary software development best practices; notably, it is poorly tested. It is built in Java, and uses Jython for rapid development of experimental procedures with access to the Java objects which represent beamline hardware. Jython is limited to python 2.7, which is no longer supported. Further, we are reaching the limits of the flexibility of data collection scripts written in plain python.

Bluesky

Bluesky [2] is a Python 3 library designed for experiment control applications. It was primarily developed at Brookhaven National Lab facility NSLS-II, and currently has contributors from a large collaboration of facilities including DLS.

It reduces the boilerplate needed to write scripts (termed 'plans') to run experiments and provides means to capture data and metadata in a variety of formats. It interfaces with the Experimental Physics and Industrial Control System (EPICS) through *Ophyd* [4] devices, essentially Python wrappers for collections of EPICS process variables with some additional convenience functions.

The *Bluesky* library provides for useful logical separation between hardware operations and data management. Plans only involve the steps which need to be taken with the beamline hardware in order to run an experiment, and data is managed through callback functions in separate modules. The main routine of *Bluesky* is termed the 'RunEngine', which consumes plans and translates them into instructions for *Ophyd* devices.

Hyperion

Hyperion is an application revolving around a set of *Bluesky* plans written to support UDC for macromolecular crystallography. Currently it includes routines for sample centring using optical images and X-ray grid scans, capturing of optical snapshots, and rotation data collection. Additional planned features include simultaneous fluorescence data collection, collection strategy determination,

† david.perl@diamond.ac.uk

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-THMBCM034

triggering of sample change robots, and a collection of various functions to manipulate beamline state so that it can run without either human interaction or GDA.

Hyperion and its documentation are published at GitHub in the Diamond Light Source organization and can be accessed at https://github.com/DiamondLightSource/*Hyperion.*

MX Data Collection and its Automation

In a typical MX experiment, a scientist first prepares a single crystal of a protein on some non-diffracting mount. This sample is then put onto a goniometer which allows orientation in the X-ray beam. The scientist aligns the sample with the X-ray beam by clicking on an optical image of the sample to identify the centre of the crystal, which is then moved to the centre of the beam. The crystal is then rotated while being exposed to the X-ray beam, and a diffraction image is captured by an X-ray detector at each angle. These images can then be used to determine the molecular structure of the protein, as well as the location and orientation of any other molecules bound to it.

At several beamlines at DLS, notably I03, we automate this process so to run continuously without human intervention. A scientist still prepares the samples on mounts, then sends them to DLS stored in liquid nitrogen, along with a desired data collection "recipe" corresponding to a type of experiment they wish to perform. The sample is then centred by means of an X-ray gridscan (as shown in Fig. 1), and a full rotation data collection is performed according to the recipe.

In order to successfully identify drug targets, several hundred or thousand chemical compounds need to be screened for their interaction with a given protein. This means collecting several hundred of these data sets, each with the crystal exposed to a different chemical compound. Therefore, we define "high throughput" as being able to collect hundreds or thousands of datasets in a reasonable time for one study. I03 processed 72230 samples in 2022, and 79555 in 2023 by the time of writing, corresponding to approximately 390 and 510 samples per day of user operations, respectively. We hope to eventually increase this figure to several thousand, allowing the screening of a full fragment library against several proteins per day.

ARCHITECTURE

Hyperion interacts with several other components of the in-house DAQ stack as well as external libraries and services. In particular, DLS is in the process of replacing its legacy Data Acquisition platform with a new service-based architecture, *Athena* [5]. Throughout this process, beamlines and users should be able to transparently continue to use the software which they have been accustomed to for over a decade. However, much of the data acquisition routines for MX are in long, procedural scripts, with many safety checks for the various pieces of beamline hardware, especially those which are moved during normal operation and could collide.

For a high throughput application, and one where we may assume the state of the beamline as it is not being modified by live users, we can disregard many of those checks if they have already been performed at some earlier step in the process. In order to transition the automated data collection at I03 from GDA to *Hyperion*, therefore, it has been necessary to rework some of the legacy software such that arbitrary sections of it can be skipped and/or delegated with external calls to *Hyperion*. This work will also allow some or all of the functionality of *Hyperion* to be extended to the other existing MX beamlines at DLS.

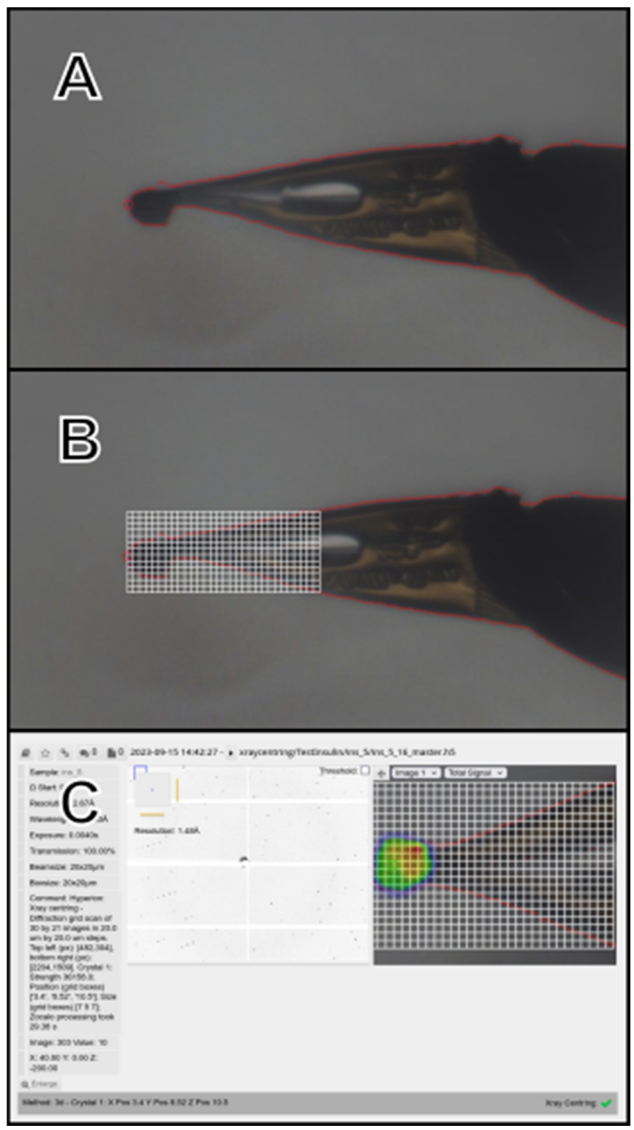

Figure 1: X-ray grid scans collected with the initial prototype of *Hyperion*. A) optical image of a crystal mounted on a loop on the goniometer, which needs to be centred, with a red outline showing the output of an edge-detection plugin used for primary optical centring. B) the same image, with an overlay showing the area to be scanned as determined in *Hyperion*. An X-ray image is collected for each point in the grid. Grid boxes measure 20 µm. C) The results of the scan as shown to users in the ISPyB web interface, with a heatmap showing the location of the crystal as determined by the X-ray signal in the collected images. Hyperion will then automatically move the centre-of-mass of this crystal into the beam for a full data collection.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-THMBCM034

Hyperion has grown out of a prototype named *Artemis* which implemented X-ray grid scans used for sample centring. This procedure is necessary to move the crystalline sample to the centre of rotation of the goniometer, so that a full diffraction dataset can be collected as the crystal is rotated in the beam. The *Atermis* prototype was used to evaluate the difficulty of implementing MX experiments using *Bluesky* and the approximate speedups we could expect, detailed further below. Figure 1 shows the results of a typical X-ray centring experiment performed using the *Hyperion* prototype.

Structure

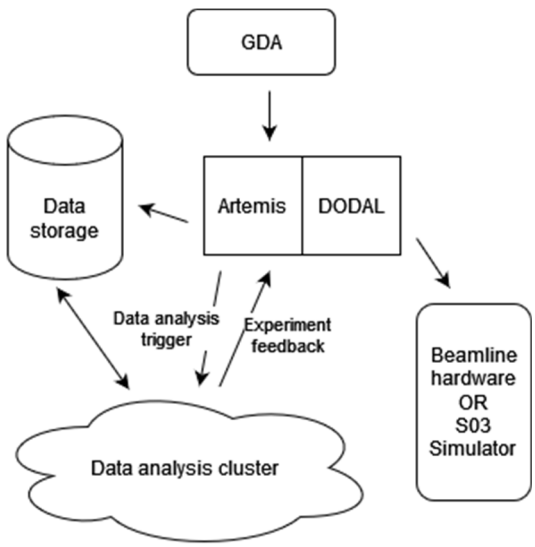

Figure 2: Structure of the initial prototype for *Hyperion*.

Figure 2 shows the architecture of the first stage prototype of *Hyperion*. Its external interface is a small *Flask* HTTP server application in the main script. In-process callbacks functions are attached to the *Bluesky* RunEngine to handle the writing of NeXuS files and the deposition of experiment metadata to the ISPyB database. This prototype has been in operation for X-ray centering for UDC at I03 since April 2023.

Currently, the *Hyperion* project is in transition from this protype to the planned architecture displayed in Fig. 3. *Hyperion* will make use of BlueAPI [6], a core component of the Athena system, which will provide HTTP server functionality. The in-process callback functions will be spun out into separate microservice processes, to improve the robustness of the main *Hyperion* process to failures to access the filesystem or database. We have already structured the NeXuS writing and database interaction modules such that they receive all of their required input in the form of experiment metadata JSON blobs emitted by the *Bluesky* RunEngine; *Bluesky* includes the functionality to dispatch these metadata through 0MQ so that moving these functions to a separate process is facile.

Device Repository: DODAL

During the development of the prototype for *Hyperion*, we found we were preparing many simple *Ophyd* device classes which would clearly be reusable between different DAQ projects and different beamlines. Therefore, the decision was made to spin these out into a separate repository, named the Diamond Ophyd Device Abstraction Layer [7] (DODAL) which is intended to contain the device logic for all hardware at DLS. DODAL gives convenient access to beamlines as python modules. These modules contain beamline profiles with preconfigured lists of hardware, and utilities for managing and instantiating these *Ophyd* device objects.

DODAL is actively developed in parallel with *Hyperion* following the principle that, in general, we wish the experiment plans to be readable by beamline scientists with general Python knowledge, such that they can verify the correctness of the experimental procedures. Therefore, complex management of device state is abstracted into the *Ophyd* device layer, improving the readability of the experiment plan scripts.

Dependencies and Support

Hyperion is primarily written in Python 3 and at the time of writing supports python versions 3.9 and up. We follow the NumPy deprecation policy [8] which is widely used in the scientific python ecosystem.

Hyperion depends on some in-house packages. The most important of these is the Diamond Ophyd Device Abstraction Layer (DODAL) [7], a repository for *Ophyd* devices used at DLS, described in more detail below.

For writing of Nexus format [9] files we use NexGen, available at https://github.com/dials/nexgen.

Experiment metatdata is deposited to an ISPyB [10] database for user access and integration with analysis pipelines.

RESULTS TO DATE

Speed-up of "Fast Grid Scans" and Unattended Data Collection

Using GDA, a typical UDC data collection takes between 110 and 120 seconds, of which around 70 are devoted to centring the crystal. Some initial increases in UDC throughput at I03 have been achieved through the simplification of the experimental procedure reducing the time by around ten seconds. A further gain of around ten seconds was achieved by arming of the detector parallel to other preparation, the implementation of which was greatly simplified thanks to the *Bluesky* RunEngine's management of asynchronous actions. This speedup represents an increase of 100 – 200 samples per day of operation.

We have implemented *OpenTelemetry* tracing of the various sections of the experimental plans in *Hyperion* to identify the slowest steps of the procedure. We intend to use the data collected from this tracing to further improve the speed of data collection.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-THMBCM034

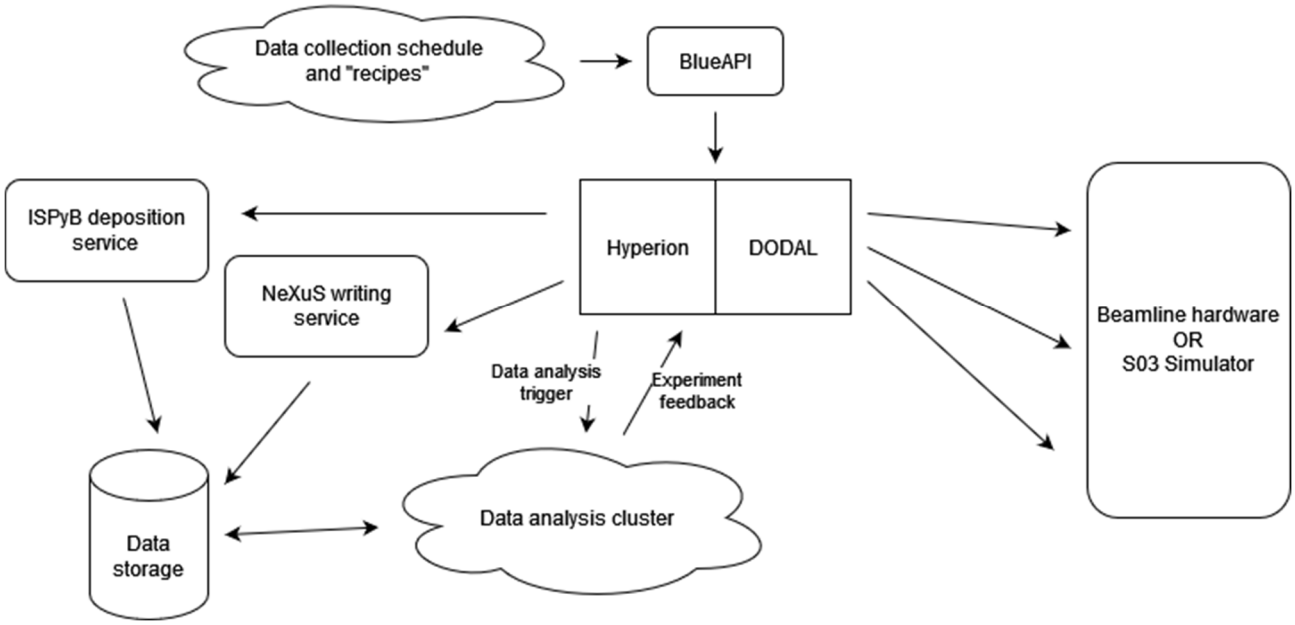

Figure 3: Planned architecture for the *Hyperion* project.

Multi-pin Data Collection

A key way in which faster UDC experiments are envisaged is through the use of "multi-pins". These are crystal mounts with several distinct samples on them. They allow the collection of multiple datasets requiring only one robotic sample loading procedure between them. In this step, a mechanical arm moves a sample from a liquid-nitrogen Dewar to the goniometer, a slow process compared to the data collection steps. Reducing the frequency of the slowest step of the UDC process will therefore greatly increase the overall throughput. Figure 4 panel A shows such a multi-pin sample. The heatmap in panel B shows where four distinct crystals have been located through a grid scan performed using *Hyperion*.

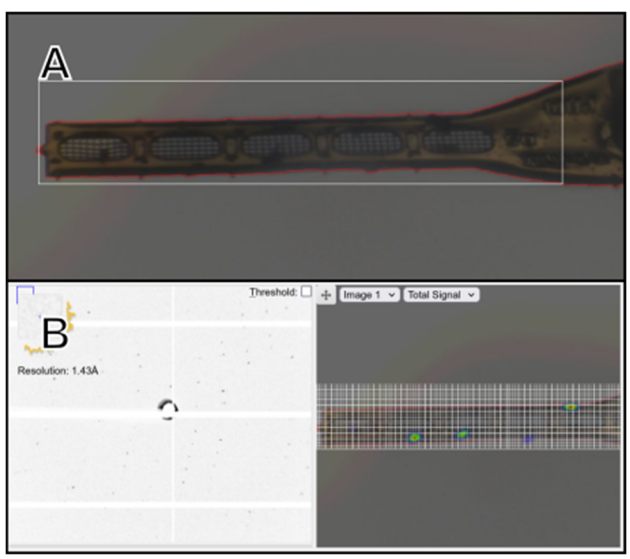

Figure 4: Screenshots of an X-ray centring grid-scan using a multi-crystal pin. A) An optical image of a multi crystal mount. The white box indicated the area to be scanned. B) The analysed data in the ISPyB web interface, with a heat map showing the locations of four individual crystals.

Improvement of Development Processes

GDA was written to support MX beamlines with similar use cases but different procedures. To reduce the maintenance burden for core use-cases, the same data collection routines were used across multiple beamlines, with conditional execution of code where requirements diverged. This naturally grew to tightly coupled, fragile code, increasing the time taken to develop improvements for experiments differing from standard, user-directed data collection. In many cases, beamline scientists have developed their own scripts to run less conventional experiments. These are therefore disconnected from the core GDA code, are often not under version control, and require extensive effort from the software team to integrate into the rest of the DAQ software stack when greater support is required.

In contrast, the flexible, modular, and hierarchical nature of *Bluesky* plans allows easy reuse of generic sections of experimental procedures. This allows for variation of procedures between beamlines with minimal code duplication. If and when beamline scientists develop new experimental plans using *Bluesky*, they can have access to the hardware as provided through DODAL instead of interacting with EPICS directly, and the results of their work will be more easily integrated into the rest of the DAQ software stack.

TESTING AND VALIDATION

Testing of experiment orchestration software is important to minimise use of valuable live synchrotron beam time for testing and prevent, at worst, damage to instruments through incorrect operations.

As a first pass, *Hyperion* is comprehensively unit-tested with approximately 90 % test coverage as measured by *Pytest*. These tests make extensive use of the *Ophyd*

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-THMBCM034

library's built-in simulation module, which is suitable for producing very simple mock devices.

Beyond this, we employ the use of a simulated beamline we call S03. The design of this simulator follows a "digital twin" philosophy, although it is currently incompletely implemented. It is composed of containerised EPICS IOCs which correspond to the real IOCs on I03, with a variety of software backends. In some cases, where these are very simple devices which do not interact with each other, this is all that is necessary.

For more complex device behaviour, we have previously employed LeWIS which provides easy means to implement behaviour and state for simulated devices such as the Eiger [11] detector. However, LeWIS does not include simple functionality for linking devices together and therefore we are developing a new simulation framework called *Tickit* [12]. This will allow us to simulate the interaction between different devices. For the MX use-case the prime example is the simulation of the Zebra [13] a signal processor which allows accurate coordination of devices. For a typical MX data collection, the Zebra can record positions of the motor controller for the goniometer while sending precisely timed triggers to the detector. *Tickit* simplifies the simulation of these interactions, with the goal of being able to run through the same experimental *Bluesky* plans in *Hyperion* in a hardware-agnostic manner.

The final validation of the correctness of the routines in *Hyperion* is the amenability of the correctness of the collected data to our standard MX data processing pipelines.

CONCLUSION

Hyperion is an under-development application for high-throughput MX data collection for current and future MX beamlines at DLS, of which the first components are already in use at I03. Currently it can process X-ray centring and rotational data collection routines, and it is intended to perform complete unattended data collection.

ACKNOWLEDGEMENTS

The authors would like to thank the whole of the MX data acquisition, data analysis, and controls teams at DLS, as well as the beamline staff involved in this project. We would also like to thank our Bluesky and Ophyd collaborators at NSLS-II and all the other facilities around the world.

REFERENCES

[1] J. Brandao-Neto and D. Hall, "Diamond-II Proposal for flagship project Beamline K04 for Ultra High Throughput XChem Fragment Screening", 2020, https://www.diamond.ac.uk/dam/jcr:bd879384-900b-4a2c-897d-a8e0e818c6e8/MX%20-%20I04-1%20XChem%20reinvention%20as%20K04%20XChem.pdf

[2] D. Allan, T. Caswell, S. Campbell, and M. Rakitin, "Bluesky's Ahead: A Multi-Facility Collaboration for an a la Carte Software Project for Data Acquisition and Management", *Synchrotron Radiat. News*, vol. 32, no. 3, pp. 19–22, May 2019. doi:10.1080/08940886.2019.1608121

[3] E. Gibbons, M. Heron, and N. Rees, "GDA and EPICS working in unison for science driven data acquisition and control at Diamond Light Source", in *Proc ICALEPCS'11*, Grenoble, France, Oct. 2011, paper TUAAUST01, pp. 529-532.

[4] K. Lauer, "ophyd Devices: Imposing Hierarchy on the Flat EPICS V3 Namespace", in *Proc. ICALEPCS'19*, New York, NY, USA, Oct. 2019, pp. 1285. doi:10.18429/JACoW-ICALEPCS2019-WEPHA083

[5] J. Shannon, C. Forrester, and K. Ralphs, 'Diamond Light Source Athena Platform', presented at the ICALEPCS'23, Cape Town, South Africa, Oct. 2023, paper TH1BCO05, this conference.

[6] BlueAPI, https://github.com/DiamondLightSource/blueapi

[7] DODAL, https://github.com/DiamondLightSource/dodal

[8] NumPy Deprecation Policy, https://numpy.org/neps/nep-0029-deprecation_policy.html

[9] M. Könnecke *et al.*, "The NeXus data format", *J. Appl. Crystallogr.*, vol. 48, no. 1, pp. 301–305, Feb. 2015. doi:10.1107/S1600576714027575

[10] S. Delagenière *et al.*, "ISPyB: an information management system for synchrotron macromolecular crystallography", *Bioinf.*, vol. 27, no. 22, pp. 3186–3192, Sep. 2011. doi:10.1093/bioinformatics/btr535

[11] I. Johnson *et al.*, "Eiger: a single-photon counting x-ray detector", *J. Instrum.*, vol. 9, no. 05, p. C05032, May 2014. doi:10.1088/1748-0221/9/05/C05032

[12] A. Emery, G. O'Donnel, C. Forrester, and T. Cobb, "Tickit: An Event-Based Multi-Device Simulation Framework", presented at the ICALEPCS'23, Cape Town, South Africa, Oct. 2023, paper TUPDP109, this conference.

[13] T. Cobb, Y. Chernousko, I. Uzun, and D. Light, "Zebra: A flexible solution for controlling scanning experiments", in *Proc. ICALEPCS'13*, San Francisco, CA, USA, Oct. 2013, paper TUPPC069, pp. 736-739.