DATA ANALYSIS SUPPORT IN KARABO AT EUROPEAN XFEL

H. Fangohr*, M. Beg, V. Bondar, D. Boukhelef, S. Brockhauser, C. Danilevski, W. Ehsan, S. G. Esenov, G. Flucke, G. Giovanetti, D. Göries, S. Hauf, B. Heisen, D. G. Hickin, D. Khakhulin, A. Klimovskaia, M. Kuster, P. M. Lang, L. Maia, L. Mekinda, T. Michelat, A. Parenti, G. Previtali, H. Santos, A. Silenzi, J. Sztuk-Dambietz, J. Szuba, M. Teichmann, K. Weger, J. Wiggins, K. Wrona, C. Xu
European XFEL GmbH, Holzkoppel 4, 22869 Schenefeld, Germany

S. Aplin, A. Barty, M. Kuhn, V. Mariani
Centre for Free Electron Laser Science, DESY, Notkestrasse 85, 22607 Hamburg, Germany

T. Kluyver
University of Southampton, SO17 1BJ Southampton, United Kingdom

Abstract

We describe the data analysis structure that is integrated into the Karabo framework to support scientific experiments and data analysis at European XFEL. The photon science experiments have a range of data analysis requirements, including online (i.e. near real-time during the actual measurement) and offline data analysis. The Karabo data analysis framework supports execution of automatic data analysis for routine tasks, supports complex experiment protocols including data analysis feedback integration to instrument control, and supports integration of external applications. The online data analysis is carried out using distributed and accelerator hardware (such as GPUs) where required to balance load and achieve near real-time data analysis throughput. Analysis routines provided by Karabo are implemented in C++ and Python, and make use of established scientific libraries. The XFEL control and analysis software team collaborates with users to integrate experiment specific analysis codes, protocols and requirements into this framework, and to make it available for the experiments and subsequent offline data analysis.

INTRODUCTION

The European X-ray Free Electron Laser (XFEL) is a facility providing X-ray and laser excited imaging to wide range of science users. The generated X-ray pulses are extremely brilliant (peak brilliance $\sim 5 \cdot 10^{33}$ photons/s/mm^2/mrad^2/0.1% band width), ultra-short (<100 fs) pulses of spatially coherent X-rays with wavelengths down to 0.1 nm. X-rays are delivered in 10 trains of pulses per second, where up to 2700 pulses form a train in which the pulse separation is 222 ns [1].

Measurement data obtained during experiments are analysed both during the experiment – to guide the experimental work during beam time – and subsequently to fully understand and exploit the data taken. The complexity of the experimental setup and high data rates of the order of 10-15 GB per second per detector at European XFEL [2] demand an efficient concurrent approach of performing experiments and data analysis: Data analysis must already start whilst data is still being acquired and initial analysis results must immediately be usable to feedback into and re-adjust the current experiment setup. The Karabo control system [3] has been developed to support these requirements.

In this paper, we describe European XFEL's provision for data analysis during and after the experiment. We start by defining rapid, online and offline data analysis, then describe the architecture of relevant frameworks at European XFEL for rapid-feedback and online analysis, offline data analysis, further user support, and close with a brief summary.

Rapid Feedback, Online and Offline Analysis

We distinguish three different data analysis scenarios:

(i) *rapid-feedback* (or near-realtime) data analysis, which we interpret as data analysis during the experiment to optimise and maintain the experiment conditions and parameters. Key requirements here are low latency in provision of data analysis results: the shorter the latency, the easier it is during the experiment to interpret and understand the effect of an intervention. For example, the adjustment of the sample position to bring it into the X-ray beam can be done the more rapidly, when integrated data analysis automatically provides the feedback if and when the beam has hit the sample. A low feedback latency is required for more effective conducting of experiments, as is well known in photon science facilities [4–9]. One aims for near real-time feedback with latency of the order of seconds or below.

(ii) *online* data analysis: data analysis to be carried out during the experiment (thus including rapid-feedback) but not requiring the same low latency. This includes analysis that is acceptable to be carried out minutes and hours after the data has been acquired.

(iii) *offline* data analysis, which summarises all remaining data analysis that takes place after the beam time has concluded. In contrast to the online data analysis, which is focused on fast and consequently often somewhat approxi-

* hans.fangohr@xfel.eu

mative feedback, the emphasis here is on the most accurate analysis and exploitation of the data that has been recorded.

RAPID-FEEDBACK AND ONLINE DATA ANALYSIS

Figure 1 outlines a schematic and simplified view of the data flow for rapid data analysis during the experiment. Starting from left, we have indicated the overall experiment which generates data through "Detectors" and "Sensors". This is processed through the "Data Management" tools of Karabo. The data can flow to the XFEL data storage (indicated as the orange-coloured house shape at the bottom) and to the "Pipeline processing".

Data Storage

Different types of data are stored: "Raw data files" store detector data without any calibration or correction applied, to allow application of different and improved calibration and correction in the future. European XFEL GmbH has a scientific data policy [10] to provide data retention for raw data for a minimum of 5 years; striving to achieve storage for 10 years where possible. We also store detector calibration data and slow control data. Slow data contains (relatively) slowly changing data such as motor and sample positions. "Metadata" describes the content of other data.

Pipeline Processing

"Pipelined processing" is a general concept of performing subsequent actions on data tokens, passed through a pipeline of algorithms via a specified data exchange format, usually without the need of intermediate file-storage. Karabo implements this concept via TCP-based p2p channels, linking individual code components, encapsulated in so called devices [3].

These devices can communicate across the network and can thus be hosted on many different computers, allowing the data flow and processing to be distributed across hardware. There are different predefined modes of data queuing and dropping if the listening device cannot keep up. Out of these building blocks, one can create large and arbitrary networks of data stream processing pipelines. Multiple pipelines can be combined to work simultaneously.

Figure 2 shows the example of a calibration pipeline which processes incoming detector data in 16 different streams for each of the 16 detector modules, carries out calibration of the data on 16 different pipelines consisting of multiple devices each (hosted on 8 different compute nodes, i.e. two pipelines on each host) in parallel. The data flow is merged back into one stream on a dedicated combiner node, before offering that data stream to be displayed in the control room or processed further by another pipeline [2].

The "Pipeline processing" box in Fig. 1 represents multiple data pipelines and thus hides internal complexity.

In Karabo pipeline processing individual data tokens contain data associated to a single train. This may be all pulses recorded by one of the MHz detectors, a digitizer, or the slow control of selected sources as relevant for analysis.

Outputs from the "Pipeline processing" may be stored in additional "Data files" at XFEL (third data storage symbol from the left in Fig. 1).

Feedback on Control GUI

The Karabo Graphical User Interface (GUI) [11] allows the creation of scenes that provide selected pieces of information to the team running the experiment. The creation and editing of scenes can be done without programming, allowing for rapid adjustment to changing experimental scenarios.

Figure 3 shows such a "Live scene" for the AGIPD detector [12] at the SPB instrument [13], imaging diffraction on a water jet. This data is offset- and relative gain corrected, after having been fed through the calibration pipelines.

Karabo Bridge

The "Karabo bridge" shown in Fig. 1 allows to connect external applications to the Karabo control and data analysis framework. In particular, "Data stream input adapter" and "Data stream output adapter" provide an interface that allows the Karabo data stream format to be converted to the requirements of other data processing applications, and to feed data back from those applications into Karabo, respectively.

Currently, the "Data stream output adapter" offers the Karabo stream through a ZeroMQ [14, 15] socket. For the rapid feedback situation, it typically uses ZeroMQ's publish-subscribe pattern, where the data sets corresponding to photon trains are offered at the same rate as they are received from the Karabo pipeline. If the connected "Data Analysis tool" does not read a data set, the data set is dropped. At the time of writing, the latency – between the detector recording the data and the data being available at the ZeroMQ interface – has been measured as about 2.5 seconds for data going through the calibration pipeline, and about 1.5 seconds without calibration. For this setup, we processed 4 trains per second, leading to a data throughput of $2.5\,\mathrm{Hz} \times 1\,\mathrm{Mpixel} \times 64\,\text{memory cells} \times 4\,\mathrm{Byte} \times 2(\text{gain} + \text{image}) = 1.2\,\mathrm{Gb/s}$.

External data analysis tools can connect to this bridge, for example through a Python client and a C++ client [16]. They can process the data stream directly, or may write the current train data into a temporary file, then load the file and process the loaded data, before repeating this for the next train data set.

OnDA [17], an open source data analysis utility designed for fast online feedback during serial X-ray diffraction and scattering experiments, has been extended to connect to the Karabo bridge and has been successfully used in the first set of experiments at XFEL.

The diagram in Fig. 1 shows that the data analysis tools that are connected through the Karabo bridge, of which there could be many, may create their own data files, and can also feed back information into the Karabo system, which can be used for feedback in the GUI and for permanent storage if required.

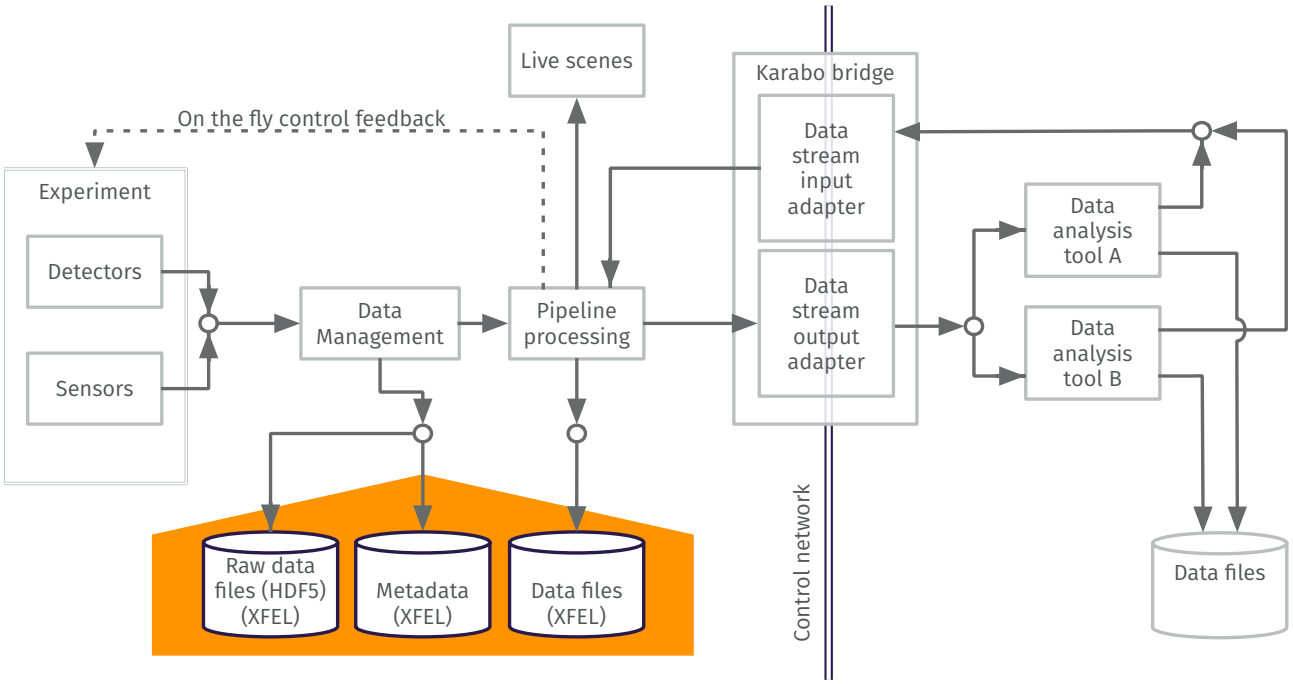

Figure 1: Schematic view of data flow for rapid feedback during experiment. See main text for a detailed discussion of all elements.

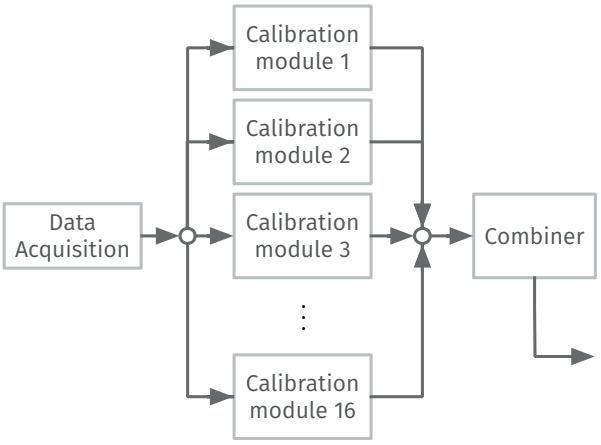

Figure 2: Simplified representation of a calibration pipeline. In this example, the data stream originating from the data acquisition is split into 16 parallel streams, so that each of the 16 calibration devices can process one of the 16 detector modules. This allows the 16 calibration devices to be distributed on up to 16 different compute nodes to spread the computational load. The data streams are subsequently united in the combiner device. Other configurations are possible.

The Karabo bridge allows data analysis tools to run on hardware and networks separated from the Karabo control system.

Complex Experimental Protocols

The combination of control and data analysis within the Karabo software allows complex experimental protocols to be supported: data analysis software can provide information based on which a scientist can manually control the experiment, but it is also possible to have *automatic control feedback* based on outcomes of complex data processing and analysis pipelines. The dashed arrow in Fig. 1 represents this real-time control feedback loop.

A challenge arising from close coupling of a wide range data analysis software is that misbehaviour of some parts of the software must not affect the control system: data analysis software crashes, memory overflows, flooding of the network, filling up disk space are possible but must be managed to reduce their potential impact. This is partly addressed by running control parts of Karabo on dedicated and locked hardware, and restricting the interaction with user provided analysis tools to that via the Karabo bridge (as sketched schematically in Fig. 1.) Only carefully reviewed software can be integrated into the control system, to ensure it cannot cause undue impact on other systems. See also [18] for a wider discussion of security.

Computing Infrastructure

The dedicated hardware to support data analysis for measurements and science at European XFEL is split into two systems. First a small computational resource (at the time of writing 140 cores distributed over 7 equal nodes with each 256 GB RAM) is set aside for each experiment during their respective beam time. These nodes are provided exclusively for the experimenting group. Additional computational facilities are available to run the control system and to provide XFEL hosted calibration and data analysis pipelines. This computer system is referred to as the “online cluster”.

Second, a larger computational resource is available for offline analysis: the high performance computing system

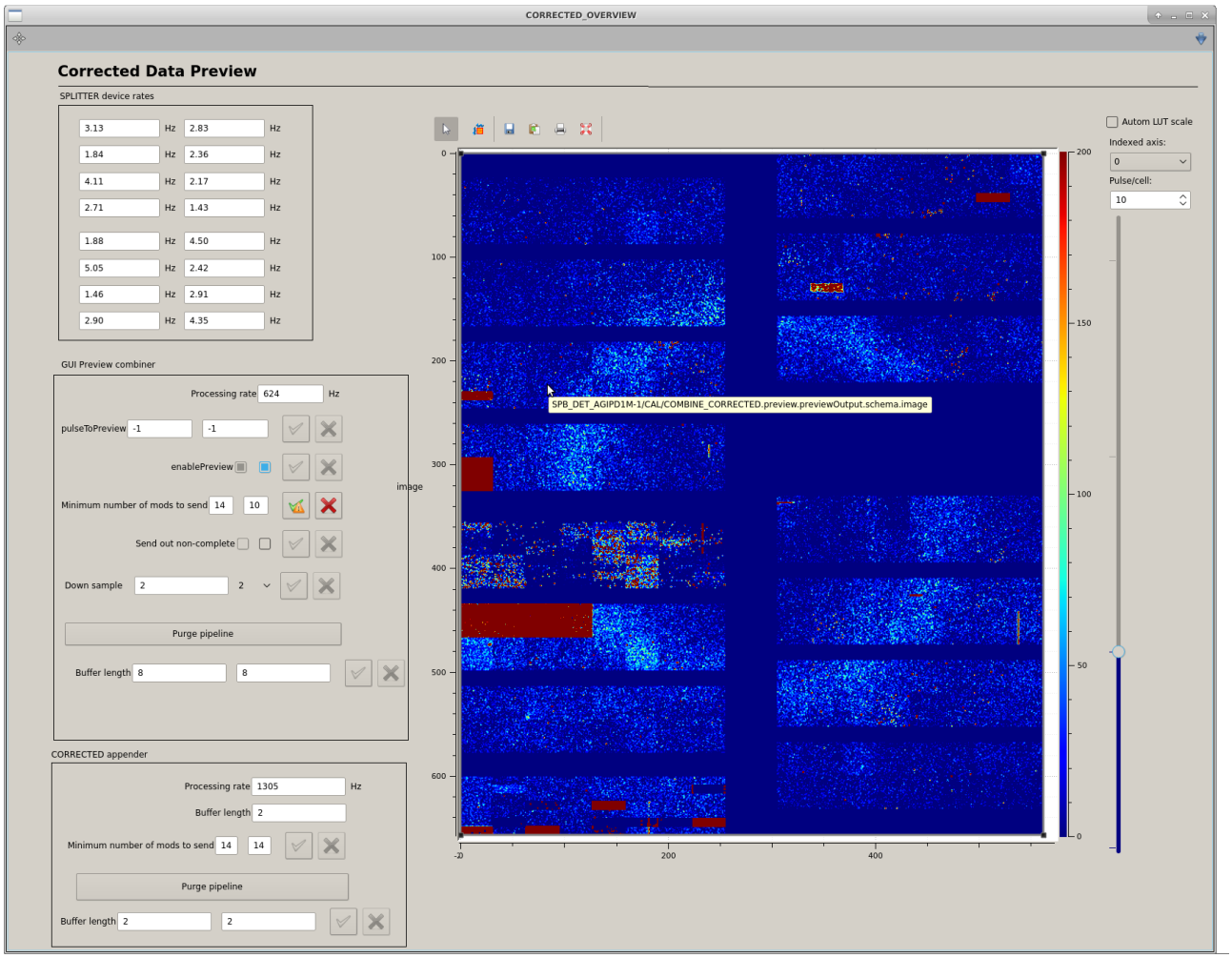

Figure 3: Rapid feedback during experiment on GUI in control room, illustrated through screenshot of GUI element that updates live and shows a single preliminary corrected image of X-ray scattering in water jet obtained with the AGIPD detector [12]. The GUI provides interactive elements, for example the slider on the right to select which pulse for a given train of pulses is to be displayed, and control and information elements on the left that affect latency and completeness of the displayed overview plot. Only 15 out of 16 detector modules are shown due to throughput optimisation.

Maxwell [19] of the Deutsches Elektronen-SYnchrotron (DESY) has been extended to provide compute power to support XFEL users during and after their experiment. Currently, there are 80 nodes available, each with 40 cores and 512 GB RAM.

The compute infrastructure has been split in this way to ensure that exclusive use of compute resources is possible and available in close proximity to the experimental hutch and data being captured by the detectors and other sensors, while the majority of compute intense analysis tasks are carried out within an existing HPC centre to benefit from the economy of scale.

Data acquired during the experiment is initially stored locally close to the experimental hutch. At the end of an experimental run (with typical run times of the order of minutes), this data is transferred to the Maxwell cluster, calibrated as appropriate, and becomes available for further offline analysis.

OFFLINE DATA ANALYSIS

Data Access

Figure 4 shows how "Preprocessed data files" in HDF5 format can be obtained for offline data analysis: on request through the metadata catalogue, a data stream is initiated, calibrated on the fly, and the resulting data stream is saved into preprocessed data files in HDF5 format. These files are made available on the Maxwell cluster, which has a reserved partition for offline data analysis. We call these *preprocessed* data files as some automatic processing may have taken place, but the main data analysis is still to come.

The particular structure of the HDF5 files will depend on the experiment carried out, but a general format is followed and examples and tools to read and process the files are available [20]. Together with users of European XFEL, we aim to create an open collection of typical data analysis recipes

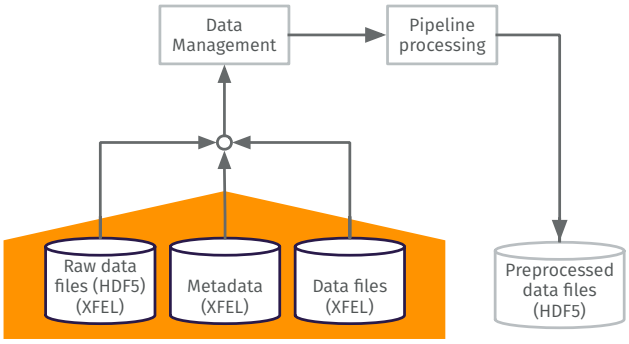

Figure 4: Recorded data can be requested from XFEL through the data management system. Raw data that is stored in European XFEL's data archive, is processed and calibrated through pipeline processing, and then delivered as HDF5 files. In this document, we call these files *preprocessed data files*. This is possible during the experiment and subsequently on demand (i.e. for online and offline data analysis).

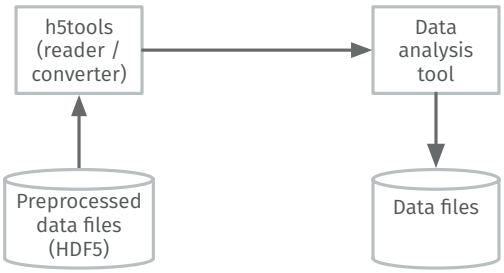

Figure 5: Preprocessed data files can be accessed directly or through file reading and converting tools provided by European XFEL. These support a growing set of standard formats, and can be used as a library to develop new custom conversions to feed the data to required data analysis packages.

over time [16]. There are two ways of further processing these files:

First, we have data file based approaches: The preprocessed HDF5 files can be interrogated, read and converted directly or through an access library and file conversion tools [16] as shown schematically in Fig. 5. Over time, we expect to respond to user requests to grow these file access utilities into general purpose tools that cover user needs, and which will provide file format conversions to existing formats that are relevant for data analysis.

Second, data stream based approaches: Fig. 6 shows that we can also replay the experiment by sending the preprocessed data files through a Karabo data pipeline and thus make the data stream available for tools connected to the Karabo bridge as described above. The advantage of this approach is that the data stream appears (from the perspective of the "Data analysis tool") as if the experiment was taking place. This allows (i) re-use of adaptors and data analysis tools that have been used for rapid-feedback data analysis

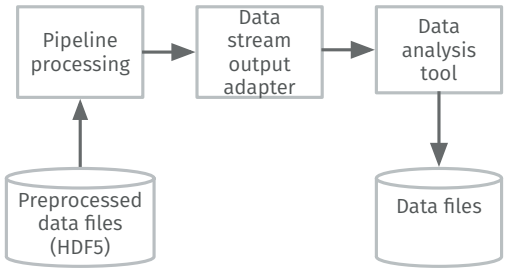

Figure 6: The preprocessed data files can also be read from disk and then sent through the Karabo data pipelines as if coming from the experiment. This data stream can then be accessed via the "Data stream output adapter" of the Karabo bridge, to connect to data analysis software that can process streaming data. This is useful to test the connection of new tools to the system in advance of allocated or desired beam time.

in the offline situation, and (ii) testing of new adaptors to connect to the Karabo bridge in advance of experiments.

For data that is streaming through the Karabo bridge in this offline analysis setting, we typically use ZeroMQ's request reply pattern, so that the data is only sent to the analysis application on request, and thus the data analysis application determines the processing rate. This is important to ensure that all data sets can be processed.

Reproducible Data Analysis – Jupyter Notebook

Documented and transparent reproducibility in computational science and data analysis has attracted increasing attention in the recent past: it is important that the procedure to obtain scientific results from data can be repeated. We integrate and support the use of the Jupyter Notebook [21, 22] to carry out data processing and data analysis, as a technology that allows straightforward documentation and sharing of data exploration and analysis, and thus supports reproducibility.

Figure 7 demonstrates basic features of the Jupyter Notebook. In essence, there is a sequence of input cells, each of which can contain text or code. Code input cells are numbered and can produce textual or multimedia output. These are displayed inside a web browser that is connected to a computational backend. This backend can be the same machine on which the notebook is displayed in the browser, or a remote server. We can see from the figure that it is possible to combine code, plots and interpretation in one Jupyter Notebook.

As notebooks can be saved, re-loaded and (assuming the relevant data files and support software is provided) also be re-executed, they help significantly in moving towards reproducible data analysis. Through a menu command or a command line interface, notebooks can be converted to widely used read-only formats such as html or pdf: the sharing of notebooks with collaborators is thus practical. The data science community in academia and enterprise has em-

braced the notebook as an executable document describing data analysis and as a high productivity tool.

While the Jupyter Notebook supports multiple programming languages, we expect that Python is likely to be used by many. The Python libraries that are available from the scientific community and from European XFEL can be combined in Jupyter Notebooks to carry out data analysis. The documentation of the detector analysis library [23] and Fig. 8 show examples.

As the computational kernel of a Jupyter Notebook is separated from the web browser that displays the notebook, this technology also provides convenient remote access to data analysis while the data and compute facilities used can remain located at European XFEL.

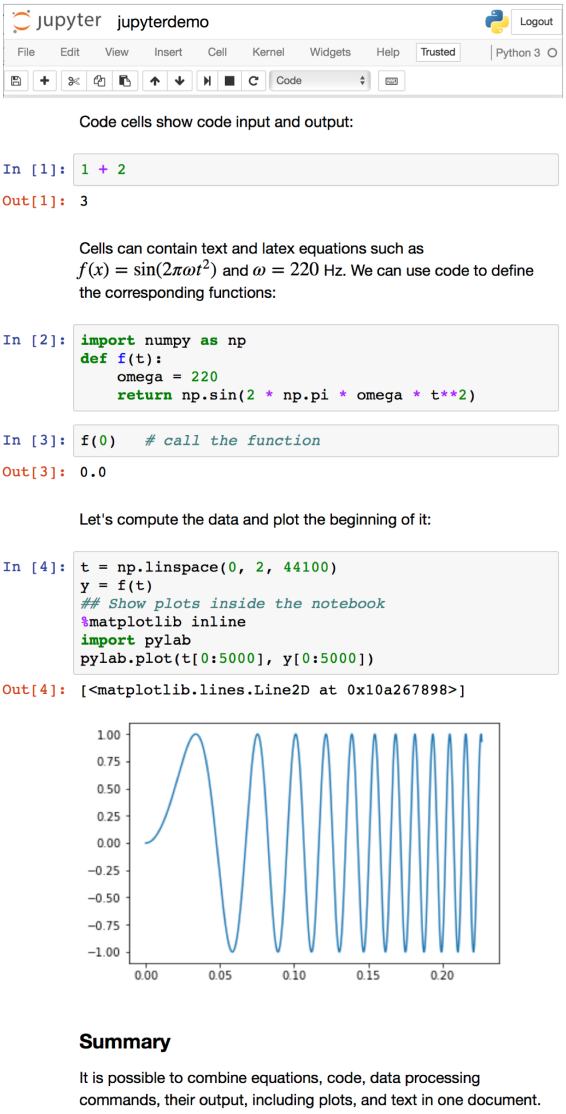

Figure 7: An example of a Jupyter Notebook, demonstrating features of value for reproducible data analysis.

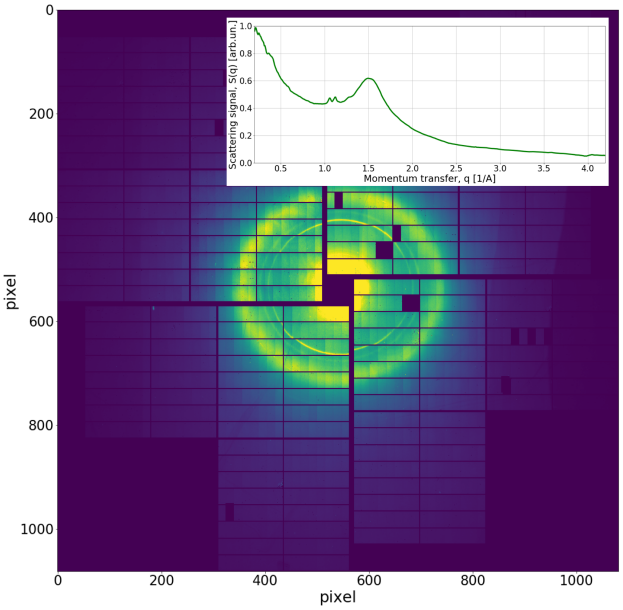

Figure 8: Liquid scattering pattern of tetrahydrofuran solution of a Cu complex collected with the LPD detector [24,25] at scientific instrument FXE [26](corrected for dark offset). Inset: Average of the azimuthally integrated set of 150 image.

USER SUPPORT FOR EXPERIMENTS

European XFEL offers a range of measures to support science users of the facility. Online documentation provides an overview of the data analysis environment [20], and provides further pointers to the documentation of the Maxwell cluster, lists of available software packages, file system and other details.

Software is made available as open source code on GitHub [16] to help with access to data files, calibration and postprocessing routines, conversion of data formats, the Karabo bridge, and to share recipes for common analysis tasks widely. We aim to develop this into a community effort with contributions from users to enable the best possible science exploitation of experiments carried out at European XFEL. Where possible, we aim to align efforts and software with other photon science facilities to avoid proliferation of new formats, protocols and software. Example data files are made available to allow users to set up and test their software in advance of their experiment.

To provide additional flexibility in providing specialist software, we are trialling the possibility to run analysis software in containers (such as Docker containers [27, 28]), which can be described as light weight virtual machines. The advantage of this model is that one can choose a particular operating system to be used within the container: this is often advantageous as research software may not have robust installation procedures and may depend on particular combinations of support libraries, and as such may depend on a particular Linux distribution or version. The containers can share the file system and/or ports with the host to facilitate data exchange.

XFEL data analysis staff are available to work with science users to provide support for data analysis, and to collaboratively develop new methods to improve tools and methods to better exploit the science hidden in the data. Experiment and data analysis requirements need to be discussed with the instrument scientists and local contacts first, and where required or desired the data analysis team will work directly with users before, during and after the beam time. On call control and data analysis support is available 24h per day.

SUMMARY

In summary, we describe the current framework for data analysis at the European XFEL facility. There is a provision of rapid-feedback, on-line and off-line analysis, including GUI elements that support rapid feedback for common experimental requirements, and an open interface protocol to connect external applications to the online and offline data stream. Data file access utilities and libraries help processing data files. A feedback loop into the experiment control is possible.

European XFEL has only just started operations and we expect significant changes and developments over time that affect and extend the framework outlined here. We look forward to working with users and other facilities to achieve the highest effectiveness and science return.

Acknowlegdements We acknowledge contributions from all staff and collaborators at European XFEL who helped shaping and realising the presented framework, and financial support from the OpenDreamKit Horizon 2020 European Research Infrastructures project #676541, and the Gordon and Betty Moore Foundation through Grant GBMF #4856, by the Alfred P. Sloan Foundation and by the Helmsley Trust.

REFERENCES

[1] M. Altarelli, R. Brinkmann, M. Chergui, W. Decking, B. Dobson, S. Düsterer, G. Grübel, W. Graeff, H. Graafsma, J. Hajdu *et al.*, "The European X-ray Free-Electron Laser," *Technical design report, DESY*, vol. 97, pp. 1–26, 2006.

[2] M. Kuster, D. Boukhelef, M. Donato, J.-S. Dambietz, S. Hauf, L. Maia, N. Raab, J. Szuba, M. Turcato, K. Wrona, and C. Youngman, "Detectors and Calibration Concept for the European XFEL," *Synchrotron Radiation News*, vol. 27, no. 4, pp. 35–38, 2014. [Online]. Available: http://dx.doi.org/10.1080/08940886.2014.930809

[3] B. Heisen, D. Boukhelef, S. Esenov, S. Hauf, I. Kozlova, L. Maia, A. Parenti, J. Szuba, K. Weger, K. Wrona *et al.*, "Karabo: An integrated software framework combining control, data management, and scientific computing tasks," in *14th International Conference on Accelerator & Large Experimental Physics Control Systems, ICALEPCS2013. San Francisco, CA*, 2013.

[4] A. Götz, M. Gerring, O. Svensson, and S. Brockhauser, "Data analysis workbench," in *11th International Conference on Accelerator & Large Experimental Physics Control Systems, ICALEPCS2011. Grenoble, France*, 2011.

[5] S. Brockhauser, K. I. White, A. A. McCarthy, and R. B. G. Ravelli, "Translation calibration of inverse-kappa goniometers in macromolecular crystallography," *Acta Crystallographica Section A*, vol. 67, no. 3, pp. 219–228, May 2011. [Online]. Available: https://doi.org/10.1107/S0108767311004831

[6] S. Brockhauser, O. Svensson, M. W. Bowler, M. Nanao, E. Gordon, R. M. F. Leal, A. Popov, M. Gerring, A. A. McCarthy, and A. Gotz, "The use of workflows in the design and implementation of complex experiments in macromolecular crystallography," *Acta Crystallographica Section D*, vol. 68, no. 8, pp. 975–984, Aug 2012. [Online]. Available: https://doi.org/10.1107/S090744491201863X

[7] S. Brockhauser, R. B. G. Ravelli, and A. A. McCarthy, "The use of a mini-κ goniometer head in macromolecular crystallography diffraction experiments," *Acta Crystallographica Section D*, vol. 69, no. 7, pp. 1241–1251, Jul 2013. [Online]. Available: https://doi.org/10.1107/S0907444913003880

[8] M. Basham, J. Filik, M. T. Wharmby, P. C. Y. Chang, B. El Kassaby, M. Gerring, J. Aishima, K. Levik, B. C. A. Pulford, I. Sikharulidze, D. Sneddon, M. Webber, S. S. Dhesi, F. Maccherozzi, O. Svensson, S. Brockhauser, G. Náray, and A. W. Ashton, "*Data Analysis WorkbeNch* (*DAWN*)," *Journal of Synchrotron Radiation*, vol. 22, no. 3, pp. 853–858, May 2015. [Online]. Available: https://doi.org/10.1107/S1600577515002283

[9] U. Zander, G. Bourenkov, A. N. Popov, D. de Sanctis, O. Svensson, A. A. McCarthy, E. Round, V. Gordeliy, C. Mueller-Dieckmann, and G. A. Leonard, "*MeshAndCollect*: an automated multi-crystal data-collection workflow for synchrotron macromolecular crystallography beamlines," *Acta Crystallographica Section D*, vol. 71, no. 11, pp. 2328–2343, Nov 2015. [Online]. Available: https://doi.org/10.1107/S1399004715017927

[10] "Scientific data policy," European XFEL GmbH, Tech. Rep., 2017, http://www.xfel.eu/users/experiment_support/policies/scientific_data_policy/index_eng.html.

[11] B. Heisen, M. Teichmann, K. Weger, and J. Wiggins, "Karabo-GUI: The Multi-Purpose Graphical Front-End for the Karabo Framework." 15th International Conference on Accelerator and Large Experimental Physics Control Systems, Melbourne (Australia), 17 Oct 2015 - 23 Oct 2015, Oct 2015, p. WEPGF153. [Online]. Available: http://bib-pubdb1.desy.de/record/295186

[12] B. Henrich, J. Becker, R. Dinapoli, P. Goettlicher, H. Graafsma, H. Hirsemann, R. Klanner, H. Krueger, R. Mazzocco, A. Mozzanica, H. Perrey, G. Potdevin, B. Schmitt, X. Shi, A. Srivastava, U. Trunk, and C. Youngman, "The adaptive gain integrating pixel detector AGIPD a detector for the european xfel," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 633, no. Supplement 1, pp. S11–S14, 5 2011.

[13] A. P. Mancuso *et al.*, "Technical design report: Scientific instrument Single Particles, Clusters and Biomolecules (SPB)," Technical Report 2013-004, European XFEL GmbH, Tech. Rep., 2013.

[14] "ZeroMQ Message Transport Protocol," https://rfc.zeromq.org//spec:23/ZMTP/, 2017, http://zeromq.org.

[15] P. Hintjens, *ZeroMQ: Messaging for Many Applications*. O'Reilley Media Inc, 2013.

[16] European XFEL GmbH, "Open source tools to support data analysis at European XFEL GmbH," https://github.com/European-XFEL, 2017.

[17] V. Mariani, A. Morgan, C. H. Yoon, T. J. Lane, T. A. White, C. O'Grady, M. Kuhn, S. Aplin, J. Koglin, A. Barty, and H. N. Chapman, "*OnDA*: online data analysis and feedback for serial X-ray imaging," *Journal of Applied Crystallography*, vol. 49, no. 3, pp. 1073–1080, Jun 2016. [Online]. Available: https://doi.org/10.1107/S1600576716007469

[18] L. Mekinda, V. Bondar, C. D. Sandor Brockhauser, W. Ehsan, S. Esenov, H. Fangohr, G. Flucke, G. Giovanetti, S. Hauf, D. G. Hickin, A. Klimovskaia, L. Maia, A. M. Thomas Michelat, A. Parenti, H. Santos, K. Weger, and C. Xu., "Securing light source SCADA systems," in *Proceedings of ICALEPCS 2017*, 2017, barcelona, 9 Oct - 13 Oct 2017, Spain, 2017.

[19] "Maxwell cluster," https://confluence.desy.de/display/IS/Maxwell, 2017.

[20] European XFEL GmbH, "Data analysis focused documentation," https://in.xfel.eu/readthedocs/docs/data-analysis-user-documentation/en/latest/, 2017.

[21] F. Pérez and B. E. Granger, "IPython: a system for interactive scientific computing," *Computing in Science and Engineering*, vol. 9, no. 3, pp. 21–29, May 2007. [Online]. Available: http://ipython.org

[22] T. Kluyver, B. Ragan-Kelley, F. Perez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, S. C. Jason Grout, P. Ivanov, D. Avila, S. Abdalla, C. Willing, and J. D. Team, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, 2016, ch. Jupyter Notebooks—a publishing format for reproducible computational workflows, pp. 87–90.

[23] S. Hauf *et al.*, "XFEL Detector tools documentation," https://in.xfel.eu/readthedocs/docs/pydetlib/en/latest/_notebooks/index.html, 2017.

[24] M. Hart, "Development of the LPD, a high dynamic range pixel detector for the European XFEL," in *Proceedings of the Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2012 IEEE*, 27. Oct. - 3. Nov. 2012, Anaheim, CA, USA, 2012.

[25] A. Koch, M. Hart, T. Nicholls, C. Angelsen, J. Coughlan, M. French, S. Hauf, M. Kuster, J. Sztuk-Dambietz, M. Turcato, G. A. Carini, M. Chollet, S. C. Herrmann, H. T. Lemke, S. Nelson, S. Song, M. Weaver, D. Zhu, A. Meents, and P. Fischer, "Performance of an LPD prototype detector at MHz frame rates under synchrotron and FEL radiation," *Journal of Instrumentation*, vol. 8, no. 11, p. C11001, 2013. [Online]. Available: http://stacks.iop.org/1748-0221/8/i=11/a=C11001

[26] C. Bressler *et al.*, "Technical design report: Scientific instrument FXE," Technical Report 2012-008, European XFEL GmbH, Tech. Rep., 2012.

[27] http://www.docker.com, 2017.

[28] C. Boettiger, "An introduction to Docker for reproducible research," *SIGOPS Oper. Syst. Rev.*, vol. 49, no. 1, pp. 71–79, Jan. 2015. [Online]. Available: http://doi.acm.org/10.1145/2723872.2723882